
ATARI
ST COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
Mai ´88
DM 7,-
Ös. 56,- Sfr. 7,-
5
61682
CeBIT'88
- Was gibt's Neues bei ATARI ?
GFA-BASIC3.0
- Noch schneller, noch besser?
SM124 MULTISYNC
- Und es geht doch !
MUSIKMESSE FRANKFURT
- Alle Neuigkeiten für den ST
ATARI
NEWS

Ein *Anblick* zum überblicken.

Allen, die nur eine oder noch gar keine Software aus unserem Hause kennen, wollten wir schon immer einmal in Kürze unsere Produktpalette zeigen. Hier sei sie beschrieben: Wir haben drei Programmiersprachen veröffentlicht, **Megamax-C** (dazu die **Editor Toolbox**), **Megamax Modula-2** und schließlich **Imagic**, die Sprache, die Bilder schreibt.

Bei Textanwendern hat sich inzwischen herumgesprochen, daß unsere Textverarbeitung **Signum!Zwei** allen Maßstäben gerecht wird, die von unterschiedlichsten Bedürfnissen diktiert werden. Mehrspaltensatz, freie Formeldefinition, Grafikeinbindung, eigener Fonteditor, Fremdsprachenzeichen sind da nur die allerwichtigsten Attribute, die wir hier nennen können. Dazu sind eine Reihe Zusatzprogramme (**Signum!Extra**) und mittlerweile über 200 verschiedene Zeichensätze (**Signum!Fonts, SiFoX**) erhältlich. Bitte ausführliche Liste anfordern.

Im Bereich der Grafikprogramme finden sich in unserer Palette zwei Schwarzweißprogramme: **STAD**, ein sehr schnelles und flexibles Zeichenprogramm mit komfortablem 3-D-Teil und (in Ankündigung) **Creator**, über das wir noch nichts verraten.

Für creative User gibt es Hilfsprogramme, die wir unter dem Decknamen **Utility Series** anbieten. Zwei Programme sind bisher erschienen: **FlexDisk** und **Harddisk Utility.**

Bolo heißt das etwas andere Ball(er)-spiel, bei dem man mit etwas Geschick und über 50 Ebenen zum Mega-Ghost gelangt. Unser erstes Spiel der **Games Series.**

Ausführliche Informationen senden wir auf Anfrage gerne zu.

APPLICATION SYSTEMS HEIDELBERG

APPLICATION SYSTEMS HEIDELBERG Englerstraße 3 Postfach 10 26 46 D-6900 Heidelberg Telefon (0 62 21) 30 00 02.





Signum!Zwei liebt STAD-Bilder.

Wo die Liebe hinfällt: Wenn nicht liebes- so ist **Signum!Zwei** doch immens g r a f i k - fähig. Natürlich ist da die V o r - liebe für **STAD**-Bilder, was wir nicht verhehlen wollen, weil wir das ja beabsichtigen.
Das heißt: Alle Bilder, die mit unserem aufwendigen Schwarz-weiß-Grafik-Programm **STAD** kreiert wurden, sind problemlos in die Texte von **Signum!Zwei** zu laden, daß es eine Freude ist. Daß es von **STAD** ein Update gegeben hat, erwähnten wir bereits an anderer Stelle: Die neue Version hat 99 Bildschirme beim Mega-ST, einen Laserdruckertreiber für den Atari-Laser und noch vieles mehr.
Wir verschicken auf Anfrage gerne ein Software-Info zum neuen **STAD,** zu **Signum!Zwei** und allen dafür erhältlichen Zeichensätzen.

Bliebe noch, auf die **Utility Series** hinzuweisen, unsere kleinen Helferlein, die flexible Ramdisk, genannt **FlexDisk** für 69,– DM und **Das Harddisk Utility,** das definitive Harddisk-Backup Programm. Ein Muß für Profis für nur 69,– DM. Software-Infos bitte anfordern.

Signum!Zwei
Die zweite Generation der umfassenden Textverarbeitung für unterschiedlichste Anwendung. Für Literaten und Musiker, für Sprachwissenschaftler, Physiker, Chemiker, Mathematiker usw.
Viele zusätzliche Fonts sind erhältlich.
448,– DM

STAD
Das Grafik-Programm, das fast keine Wünsche offen läßt. Die neue Version: 99 Bildschirme beim Mega-ST. Laserdruckertreiber für Atari-Laser.
179,– DM



APPLICATION SYSTEMS HEIDELBERG Englerstraße 3 D-6900 Heidelberg Telefon (0 62 21) 30 00 02.

EDITORIAL

Das erste Mal

Jetzt bin ich also zum ersten Mal auf der ersten Seite einer ST-Computer; auf der Seite, die in der Zählweise der Zeitungsmacher ja schon Seite 3 ist. Grund für mein Erscheinen hier ist die Messe aller Messen - die CeBIT'88. Nicht, daß ich zum ersten Mal auf der CeBIT-Messe war - nein, zum ersten Mal war ich aber nicht nur Schauender, sondern auch Beschauter. Zumindest konnten wir uns auf unserem etwas abgelegenen Stand der Merlin Computer in Halle 17/Stand A 70 nicht über mangelnde Aufmerksamkeit beklagen. Sie als Leser dieser Zeitschrift und ATARI-Anwender erschienen zahlreich, um Kritik und Lob wohlverteilt zu geben. Harsche Kritik teilweise, die sich über Layout und Gestaltung, seltener über den Inhalt ausließ. Und Lob und Anregungen, die wiederum die Kritik leichter ertragen ließen. Nicht, daß wir uns der Kritik erwehren wollten, sie war und ist ja berechtigt und gewünscht. Wir gelobten Besserung, doch lassen Sie uns etwas Zeit; gut Ding will Weile haben und solch eine Zeitschrift hat einen langen Vorlauf.

Für mich war es das erste Mal, daß ich Kritik zu einem Artikel einstecken mußte - ungewohnt zumindest, aber auch erklärbar, da es mein erster Artikel überhaupt war (Grundlagenserie: Die Festplatte). Dazu jedoch mehr im Anhang zu Teil IV in diesem Heft.

Ich hatte auf dieser CeBIT auch zum ersten Mal ein kleines Kärtchen, auf dem in grünen Lettern das Wort PRESSE zu lesen war. Mit diesem Sesam-öffne-dich bekamen wir viele Sachen zu sehen, die wir in dieser Ausgabe im Messebericht vorstellen möchten. Auch bei ATARI gab es viele Sachen zu sehen, die wir in dieser Ausgabe im Messebericht vorstellen möchten - auch wenn einiges nie erscheinen wird. Auf der ATARI-Pressekonferenz habe ich zum ersten Mal den quirligen Jack Tramiel erlebt - als erzürnten Racheengel. Grund für diesen Zorn ist die amerikanische Drohung an Japan, Einfuhrbeschränkungen für dynamische Speicherchips aufzuerlegen, falls Japan die Ausfuhr in die Staaten nicht drosselt. Grund genug für Herrn Tramiel, nun seinerseits die Regierung unter Druck zu setzen - was, wie er selber zugibt, eine lange Zeit in Anspruch nehmen wird. Notfalls würde er ein Halbleiterwerk kaufen, auch auf die Gefahr hin, daß dadurch sein Reingewinn von 57 Mio. Dollar auf 56 Mio. Dollar schrumpfen würde (Originalton Tramiel!). In diesem Sinne: viel Spaß mit unserem Messestreifzug und bis zum - äh... nächsten Mal.

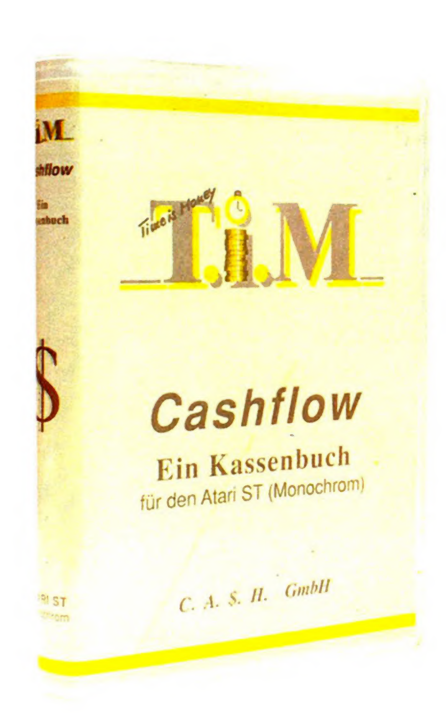
hp

I N H

SOFTWARE

HARDWARE

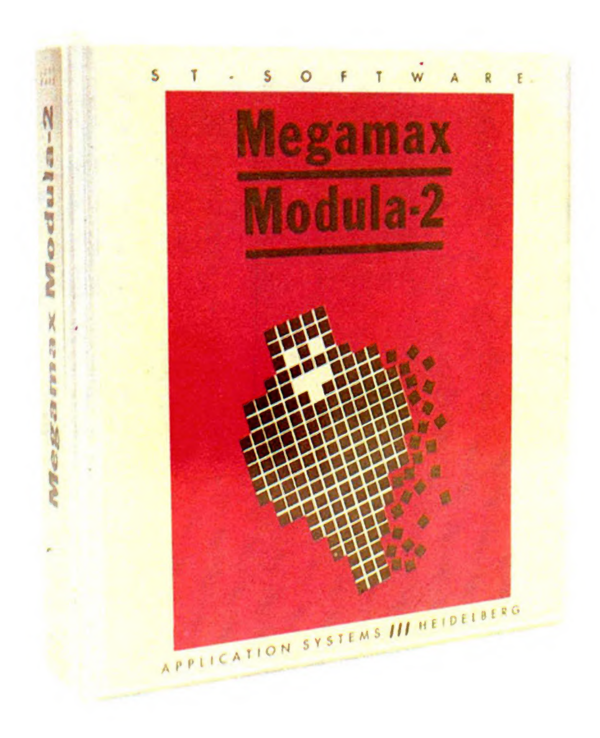
ANWENDUNGEN



Junior Prommer - die Software

Einige unter unseren Lesern werden vielleicht in der letzten Ausgabe die Software zu unserem Hardwareprojekt Junior Prommer vermißt haben. Hier ist sie nun mit einer Ausgabe Verspätung. Aufgelistet sind alle wichtigen Routinen, die zur Programmierung von EPROMs der 27xxx-Serie notwendig sind.

Seite 50



Megamax Modula-2

Viele Leser werden sagen, schon wieder ein Modula-Test. Doch dieser Pascal-Nachfolger (Modula-2 wurde ebenfalls von Herrn Wirth kreiert) findet immer mehr Freunde unter den ATARI ST-Besitzern. Megamax Modula-2 ist nun nach langer Vorankündigung (Programmierer leiden manchmal an einer Art Perfektionswahn bzw. “das muß noch rein und dies muß noch optimiert werden”) endlich auf den Markt gekommen. Wir berichten hier ausführlich über das neue Modula-2-Entwicklungssystem von Application Systems /// Heidelberg.

Seite 100

A L T

GFA-BASIC 3.0

Anläßlich einer Pressekonferenz im Hause GFA Systemtechnik wurde am 30.3.88 den anwesenden Redakteuren eine Vorabversion des neuen GFA-BASIC 3.0 überreicht. Wir machten uns sofort daran diese Version unter die Lupe zu nehmen. Was dabei herauskam, können Sie in unserem Bericht ab Seite **40** lesen.



CeBIT '88

Und wieder ist eine Messe vorbei! Man fährt zurück nach Hau..., STOP !!! Das gilt nur für die Aussteller. Wir müssen leider in die Redaktion, um auch Sie, liebe Leser, zu informieren, was es alles Neues von und um den ATARI ST zu sehen gab. 68030-Rechner, CD-ROM, Wechselplatte, Transputer, neue Hard- und Software, mehr darüber kann man in unserem Messebericht ab Seite **8** lesen.

HANNOVER MESSE
CeBIT '88
Welt-Centrum Büro · Information · Telekommunikation
16. – 23. MÄRZ 1988

GRUNDLAGEN

PROGRAMMIERPRAXIS

AKTUELLES

RUBRIKEN

NEWS

FEHLERTEUFEL BEI HDplus-News

Leider hatte sich in der letzten Ausgabe ein Fehlerteufel eingeschlichen. Dort war angegeben, daß die Festplatte HDplus 20 der Firma Vortex auf DM 1138,- gesenkt wurde. Dies ist leider falsch, da es sich hier um den Nettopreis handelt. Will man solch eine Festplatte erwerben, muß man immer noch DM 1298,- (unverbindliche Preisempfehlung des Herstellers) bezahlen. Wir bitten dies zu entschuldigen.

Preissenkung bei Bavaria-Soft

Die Bavaria-Soft Datentechnik GmbH hat die Preise der ATARI ST Programme BS-Handel, BS-Fibu und BS-Timeadress um bis zu 50% gesenkt! Mit der Fertigstellung des neuen BSS-PLUS Systems will Bavaria-Soft den ST-Anwendern mit dieser Preissenkung eine echte "Low-Cost"-Alternative für etwas über 1000 DM bieten.
Der neue Preis für BS-Handel beträgt DM 498,-, für BS-Fibu DM 598,- und für BS-Timeadress DM 149,-. Alle Anwender, die eines dieser Programme ab dem 01.01.88 gekauft haben, erhalten die demnächst erscheinenden Update-Versionen 3.0 als Ausgleich umsonst.

Bavaria-Soft Datentechnik
Otto-Hahn-Str. 25
8012 Ottobrunn
Tel.: 089/6097838

Banktransfer

Ende April erscheint mit Banktransfer das dritte Produkt der Time is Money-Serie für den ATARI ST auf dem Markt. Es handelt sich dabei um eine moderne Verwaltung für Zahlungsvordrucke. 9 verschiedene Zahlungsträgerformulare lassen sich jeweils durch die Selektion einer Adresse, Angabe des Kontos, Betrag und Verwendungszweck auf einfachste Weise zu einem Auftrag verbinden und auf die entsprechenden Endlosformulare drucken. Es werden auch Dauer- und Sammelaufträge erfaßt und verwaltet.
Abgeschlossene Aufträge werden als offene Posten behandelt und können in Zusammenarbeit mit dem Buchführungsprogramm TiM 1.1 als Buchungssätze verwendet werden. Die integrierte Adressverwaltung (mindestens 1000 Adressen) verwaltet die Empfänger und Auftraggeber von Zahlungsaufträgen. Unverbindliche Preisempfehlung: DM 298,-, Demodiskette: DM 10,-, Handbuch DM 30,-. Das Handbuch wird beim Direktkauf von Banktransfer angerechnet.

C.A.$.H GmbH
Robert-Bosche-Str. 20a
8900 Augsburg
Tel.: 0821/703856

PD-Aktion für Amnesty

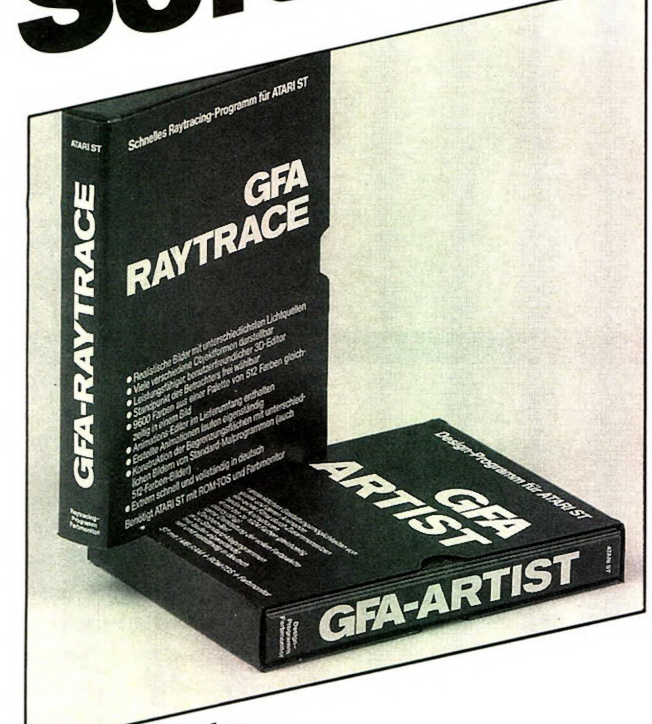
Eine in der Branche noch nicht dagewesene Public Domain Aktion startet die Firma SALIX. Sie bietet 3 PD-Disketten für je DM 15,- an, die in SALIX PROLOG 2 geschriebene Programme aus verschiedenen Sachgebieten enthalten. Von diesen DM 15,- gehen pro Diskette DM 10,- an die Gefangenenhilfsorganisation Amnesty International. Es fallen keine zusätzlichen Versandspesen an. Eine Liste der angebotenen Software ist unter folgender Adresse erhältlich:

SALIX Systeme
Mühlhäuserstr. 10
6129 Lützelbach 5

Achtung Scheinfirma!

Eine Scheinfirma namens SYNELEC GmbH mit einem Absender in Wiener-Neustadt in Österreich versendet zur Zeit Rundbriefe mit betrügerischem bzw. erpresserischem Inhalt.
Obengenannte Firma verschickt Briefe an Computerbesitzer, mit der Beschuldigung, gewerblich mit Raubkopien zu handeln. Dieser Sachverhalt sei einwandfrei und jederzeit zu beweisen. Die Rechtsanwälte beziffern den auf diese Weise der "Firma" entstandenen Schaden auf 70000 ÖS (=10000 SFr / 10000 DM). In dem Schreiben wird ein Gerichtsverfahren angedroht, wenn nicht innerhalb einer Woche eine Unterlassungserklärung mitsamt "100 DM Beweisführungsgebühr (!)" an die "Firma" gesendet werden. Dem Schreiben liegt eine "Belehrung des Rechtsanwalts" bei.
Weder der Brief noch das Schreiben der Rechtsanwälte haben einen Briefkopf, eine Telefonnummer oder eine richtige Adresse. Angegeben wird nur eine postlagernde Adresse und eine Kontonummer, an die das Geld überwiesen werden soll. Ferner ist das Schreiben eine einfache Fotokopie mit nachträglich eingedruckter Adresse.

Software-Leistung



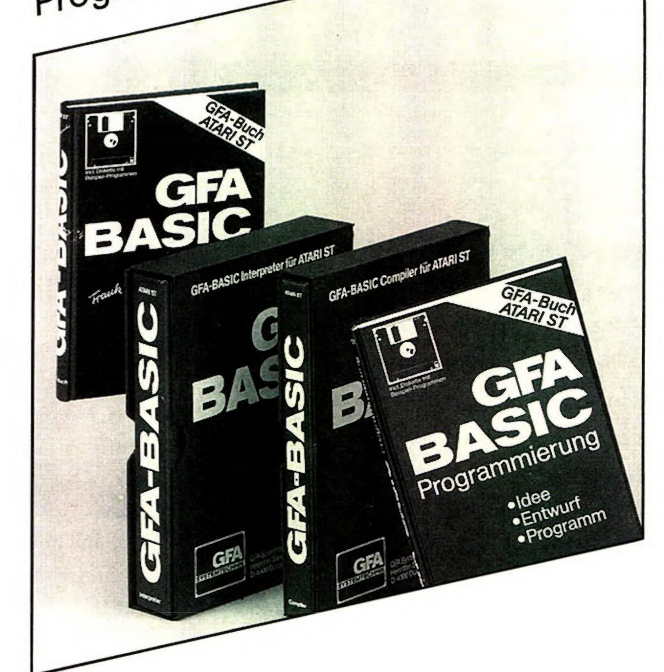
Animation in Farbe

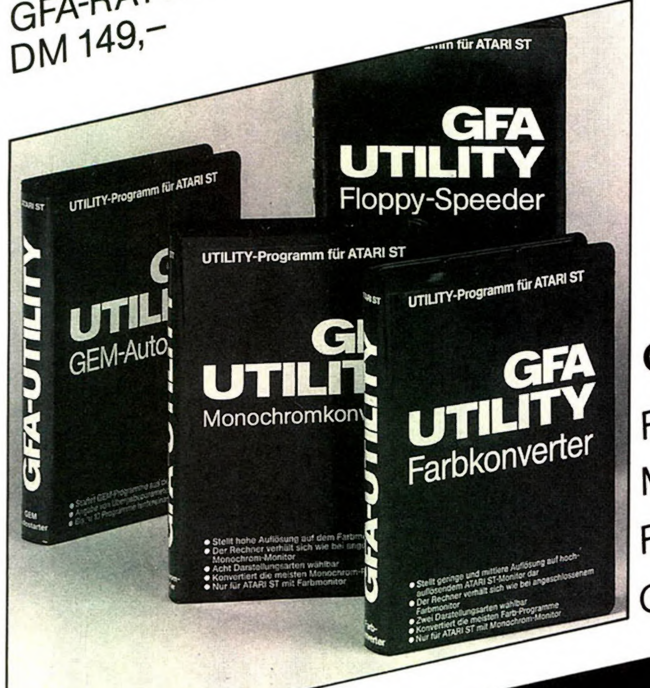
GFA-ARTIST
DM 149,–

GFA-RAYTRACE
DM 149,–

GFA-BASIC 2.0

GFA-BASIC Interpreter 2.0	DM 99,–
GFA-BASIC Compiler 2.0	DM 99,–
GFA-BASIC 2.0 Buch	DM 79,–
GFA-BASIC 2.0 Programmierung	DM 49,–





GFA-Utility-Serie

Farbkonverter	DM 59,–
Monochrom-Konverter	DM 59,–
Floppy-Speeder	DM 59,–
GEM-Autostarter	DM 59,–

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 02 11/58 80 11



CeBIT 88

Stand die letzte CeBIT im Zeichen des Schnees, so war es diesmal der Regen, der für schmutzige Schuhe sorgte. Ein dunkler düsterer Himmel, Hochwasser überall und kalte unangenehme Regenschauer waren diesmal unsere ständigen Begleiter in Hannover.

Trotz nasser Witterung wurden die Besucherzahlen vom letzten Jahr wieder übertroffen. Mit über 480.000 Besucher ist die CeBIT in ihrer Bedeutung bestätigt.

Hubert-H. Lange, Messe-Vorstandsmitglied, beschreibt die CeBIT'88 mit einem einzigen Satz: “Die höchsten Erwartungen wurden weit übertroffen”. Diese Stellungnahme läßt wohl keine Fragen mehr offen. Mit über 480.000 Besuchern übertraf die CeBIT '88 um fast ein Drittel die Zahl der Vorjahresmesse. Im dritten Jahr nach der Trennung der Hannover Messe (Industrie und CeBIT) sind die Besucherzahlen um ein Fünftel gewachsen. Erfreulich war auch die hohe Zahl der ausländischen Besucher, mit 97000 lag auch diese Zahl weit höher als im Jahr davor (69000). 17500 kamen allein aus Übersee.

Im Mittelpunkt stand dieses Jahr der Anwender und die Problemlösung mittels eines Rechners. In den Messehallen 5, 6 und 7 wurde ein besonders starker Besucherandrang festgestellt. Hier vereinten sich hauptsächlich computergestützte Branchenlösungen für den Klein- und Mittelbetrieb. Gerade in diesem Bereich hat die Computerindustrie in der letzten Zeit, wahrscheinlich wegen der sinkenden Preise der Hardware, eine sehr solide und ständig zunehmende Kundschaft. Die Schwerpunkte waren hier die Systemlösung für die Handels- und Handwerksbranche. Veranstaltungen, wie das “CeBIT Forum 88” oder das Computer-Camp waren immer ein attraktiver Treffpunkt für Computerenthusiasten.

ES GEHT VORAN!

Alle Jahre wieder, oder wie man 1988 die Ankündigungen von 1987 bestaunen kann.

Wie jedes Jahr hatten sich unter dem Dach von ATARI verschiedene Softwarehäuser, Hardwarehersteller sowie Distributoren zu einer bunten Palette vereint. Nach drei Jahren kann man auf dem Markt eine gewisse Müdigkeit spüren und die Aussteller sind mit ihren Ankündigungen etwas vorsichtiger geworden. Geblieben ist in jedem Fall, daß fast alles, was gezeigt wird, nicht unbedingt ein neues Produkt ist. Das meiste wurde bereits im Herbst (auf der Systems in München) oder ein Jahr davor, am gleichen Platz und auf der gleichen Messe, im allerersten Anfangsstadium vorgestellt. ATARI selbst ist in diesem Bereich keine Ausnahme. Was an Neuigkeiten zu sehen war, oder was vorher angekündigt und in Hannover fertig war, haben wir aus Platzgründen in zwei Teile geteilt. Wir werden in diesem Heft alle von ATARI vorgestellten neue Produkte beschreiben, und im nächsten Heft werden wir über die neue Soft- sowie Hardware, die von verschiedenen Herstellern ausgestellt wurde, berichten.

Eine Viertelmillion ATARI ST verkauft und..........

Mittwoch abend im Maritim Hotel ist ein wichtiger Termin für alle Redaktionen, die mit ATARI Computern zu tun haben. Denn dann lädt ATARI, trotz Europa-Pokal, (spielt nicht dieses Jahr wieder eine deutsche Mannschaft, die das Commodore Embleme auf ihrem Trikot trägt?) zu seiner Pressekonferenz.

“... wir haben es wieder geschafft, einige attraktive Fußballspiele zur selben Zeit wie unsere Pressekonferenz stattfinden zu lassen.” So begann Alwin Stumpf mit der überaus positiven Bilanz des Jahres 1987. Die deutsche ATARI GmbH



ist ohne Zweifel die erfolgreichste und größte außeramerikanischer ATARI-Gesellschaft. Weiterhin ist Europa selbst mit Sicherheit das wichtigste Marktsegment überhaupt. Daß es eine außerordentlich wichtige Rolle spielt, zeigt die Gründung zweier neuer Tochtergesellschaften: ATARI-Spanien und ATARI-Skandinavien.

Alwin Stumpf konnte an diesem Abend über den andauernden erfolgreichen Kurs ATARI Deutschlands berichten, der sich mit einem Umsatzanstieg von mehr als 25 Prozent gegenüber dem Vorjahr bestätigt hat. Um der Bedeutung des deutschen Anteils gerecht zu werden, muß man sagen, daß ATARI Deutschland mit 175 Mio. DM zu etwa 21 Prozent zum weltweiten Umsatz der ATARI Corporation beiträgt.

Lieferengpässe und Verzögerungen

Trotz einer Steigerung der Produktionskapazität 1987 (es wurden weltweit seit August drei Schichten gefahren) hat ATARI wieder Lieferschwierigkeiten gehabt. Insofern wurden für dieses Jahr in erster Linie die Erweiterung der Produktionskapazitäten angesetzt. Nach Aussage von Alwin Stumpf hätte ATARI Deutschland wesentlich mehr verkaufen können, wenn nachgeliefert hätte werden können.

ATARI Deutschland ist eine ST- Gesellschaft

Bei 250.000 verkauften Geräten kann man wohl sagen, daß der Schwerpunkt im ST Bereich liegt. Das gleiche Verhältnis zeigt sich in Frankreich, Holland, Belgien und in der Schweiz. Anders als in England und Amerika selbst, wo ungefähr 60 Prozent des Umsatzes im Consumer-Bereich liegt. Konsequent zu dieser Entwicklung wird ATARI Deutschland die ST-Computer-Linie weiter ausbauen und verbessern.

ATARI-Technologiezentrum in Braunschweig

Um der Wichtigkeit der ATARI-Produkte in Europa gerecht zu werden, wird noch in diesem Jahr ein europäisches Entwicklungszentrum in Braunschweig gebaut. Dieses Zentrum wird in Zukunft die speziellen Bedürfnisse des europäischen Marktes befriedigen. ATARI-Deutschland spielt auch mit dem Gedanken einige Hardwareentwicklungen in diesem Zentrum zu verwirklichen. Braunschweig wurde mit Sicherheit deswegen gewählt, weil in vielen Universitäten, die in Norddeutschland ansässig sind, immer mehr am ATARI ST entwickelt wird.

Es ist zu erwarten, daß in Zukunft die neuen Produkte von ATARI nicht nur aus dem sonnige Sunnyvale kommen werden, sondern auch aus dem Technologiezentrum in Braunschweig.

ATARI Deutschland steigt verstärkt in den Softwarevertrieb ein

Die Anzahl der Software, die direkt von ATARI angeboten wird, wird auch in diesem Jahr größer. Es handelt sich in der Regel um fremde Produkte, dessen Vertrieb von ATARI übernommen wird. Für die Übernahme des Vertriebs von Software wird ATARI, je nach Fall verschiedene Modi haben. Entweder wird das Produkt ganz übernommen (ATARI hat als einziger das Vertriebsrecht) oder ein Mischverfahren, in dem Hersteller und ATARI das Produkt vertreiben werden. Das ist zum Beispiel der Fall mit Produkten wie "Calamus", dem DTP-Software-Paket der Firma DMC, daß in einer Einsteigerversion von ATARI angeboten wird, und in einer noch leistungfähigeren vom Hersteller selbst.

Aber selbst die Produktion eigener Software wird gefordert. So wird direkt nach der Messe eine Computer-Tutor-Software auf den Markt gebracht, die eine hundertprozentige ATARI Produktion darstellt.

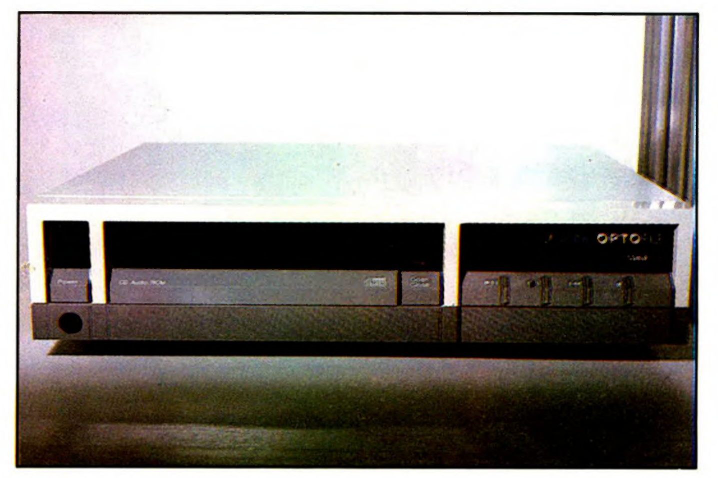
Alte Probleme und neue Technologien

Eines der alten, noch nicht gelösten Probleme von ATARI ist die Lieferung größerer Stückzahlen des Blitter-Chips. Auch hier wurde bei der Pressekonferenz angekündigt, daß in nur wenigen Wochen ausreichende Mengen an Chips ausgeliefert werden sollen. Wenn alles klar über die Bühne geht wird ATARI mit der Produktion von Blitter-Chips im DIL-Gehäuse anfangen.

DATEN VOM LASER

"OPTOFILE CDAR504" protzt es vom Gehäuse der CD-ROMs von ATARI. Für 1198 DM kann man sich das Vergnügen leisten, während der Arbeit am Rechner klirrfrei und entsetzlich dynamisch Dire Straits zu hören, und im nächsten Moment schon im Englisch-Deutsch-Lexikon auf CD zu blättern, um herauszufinden, was dieses Dire Straits denn eigentlich auf deutsch bedeutet (oder um sich die Bedienungsanleitung für das brandneue CD-ROM, Wort für Wort selbst ins Germanische zu übertragen).

Ernst beiseite: Nach langen Ankündigungen hat ATARI anscheinend doch noch einen Hersteller gefunden, mit dem das Motto "Power without the price" auch bei CD-ROMs zu verwirklichen war. Zu sehen war das CD-ROM aber auf der Messe nur am letzten Tag.



Das CD-ROM

An jenem Tag kam die Software aus Sunnyvale am Messestand an, so daß man eine Datenbank auf CD-ROM bewundern durfte.

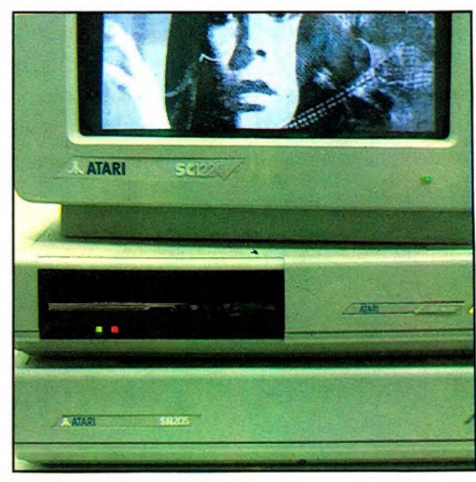
Gegenüber der Demonstration auf der Comdex hatte sich aber nicht viel geändert. Laut ATARI ist die Software mittlerweile nur soweit verbessert worden, daß sie auch wirklich nicht nur als Demoversion läuft. Als möglicher Auslieferungstermin wurde uns der Herbst genannt (der Herbst kann aber sehr lang werden).

BÄUMCHEN WECHSEL DICH...

Mit der Ankündigung einer Wechselplatte hat ATARI wieder einmal alle überrascht. Zwar war der Prototyp nur in den geheimen Gemächern des CeBIT-Standes zu bestaunen, aber immerhin gab es vorab schon einige technische Daten, auf daß den ST-Besitzern das Wasser im Munde zusammenlaufe. Et voilà:

Das Ding, getauft auf den prosaischen Namen SR444, bietet auf einer einzigen Wechselkassette Platz für knapp 43 Megabytes (formatiert; unformatiert fast 54 Megabytes). Jede von diesen Kassetten soll (im Moment) 120 US-$ kosten, also schätzungsweise unter 300 Märker. In welchen Preisregionen sich das Gerät selbst bewegen wird, steht in den Sternen (Richtung Krebsnebel, würde ich sagen).

Auf 1278 Zylindern werden jeweils 68 Sektoren á 512 Bytes untergebracht; damit kommt man auf die oben genannten 43 Megabytes (1278*68*512 = 43452 KB). Zum



Die 43 MB Wechselplatte

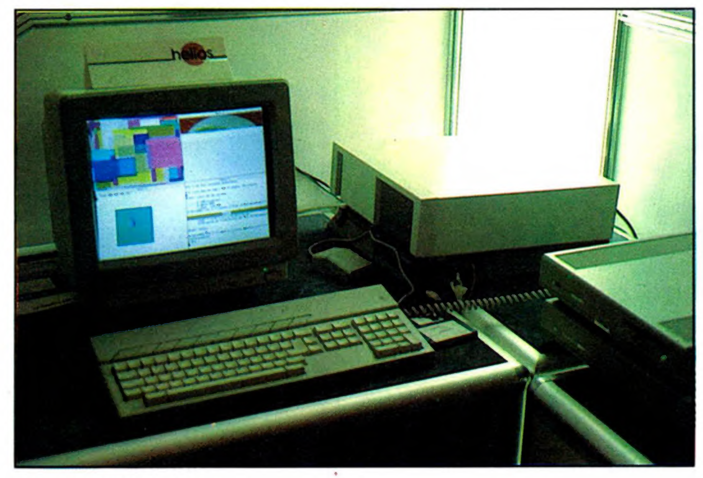
Vergleich: Bei SH204 und SH205, den Standard-Festplatten von ATARI, werden nur 612 Zylinder (maximal etwa 660) formatiert.

Wirklich angenehm ist die niedrige mittlere Zugriffszeit von 25 Millisekunden (SH205: 85 ms).

Wenn Sie jetzt schon Regenmantel und/oder Turnschuhe angelegt haben, um zu Ihrem Händler zu huschen: Viel zu früh. Wahrscheinlich können Sie sich frühestens nächstes Jahr über dieses schmucke Stück freuen.

ABAQ - Die ATARI-Transputer-Workstation

Auf der CeBIT war das erste Mal in Deutschland ATARIs neues Wunderkind, die ABAQ Transputermaschine zu sehen. In diesem Gerät verbirgt sich die Leistung einer modernen Workstation zum Preis eines größeren PCs. Dieser Artikel verrät Details.



Die ABAQ-Workstation

Zuerst einmal, bevor wir uns weiter ins Dickicht der Details begeben, was ist eigentlich eine Workstation? Unter 'Workstation' versteht man Rechner, die ungefähr (so mehr oder weniger halt) folgende Forderungen erfüllen:

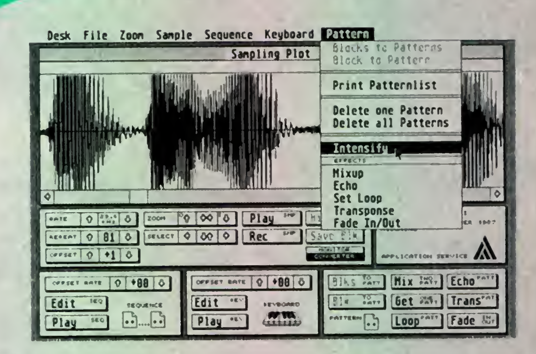
* Mindestens 1 Millionen Instruktionen pro Sekunde Rechenleistung
* Mindestens 1 MByte Hauptspeicher
* Hochauflösende Grafik mit mindestens 1 Million Bildpunkte
* Schneller Netzwerkanschluß mit mindestens 1 MByte/Sec Durchsatz
* Multitasking-Betriebssystem, am besten mit Fenstertechnik

Eine typische Workstation, z.B die Apollo Domain DN 4000, enthält einen 68020 Prozessor mit Arithmetik-Coprozessor, typisch mit 16-25 MHz Taktfrequenz. Das ganze läuft unter Unix oder einem ähnlichen Betriebssystem und kann über eine Ethernet-Schnittstelle mit bis zu 1,5 Megabyte pro Sekunde kommunizieren.

Bisherige Workstations, die diese Forderungen erfüllen, kosten ab 40.000 DM. Das ATARI-Transputersystem erfüllt alle Forderungen an eine Workstation und wird unter 10.000 DM kosten, übertrifft mit seiner Leistung aber auch Workstations einer höheren Preisklasse. Seine Fähigkeiten erhält das ATARI Transputer-System durch einen Chip der Firma Inmos, den T800-Transputer. Ein Transputer ist im Prinzip ein gewöhnlicher 32-Bit Mikroprozessor, der allerdings besonders schnell ist und einige ausgefallene Fähigkeiten besitzt.

Transputer sind RISC-Prozessoren. RISC ist eine Abkürzung für 'Reduced Instruction Set Computer', was 'Computer mit eingeschränktem Befehlssatz' bedeutet. Diese Technik gewinnt zunehmend an Bedeutung, weil man Maschinen mit kleinem Befehlssatz erheblich schneller bauen kann, als solche mit einem großen





Die AS SOUND SAMPLER

Das G DATA Sound Sampling System für alle ATARI ST Computer. Jetzt in neuem Gewand mit neuem Konzept. Für jeden Anwendungszweck und Geldbeutel das richtige.
Die AS SOUND SAMPLER bieten brillanten Sound voller Dynamik und Leistung. Sampeln unter der leicht verständlichen Benutzeroberfläche ist sampeln total. Samples aufnehmen, editieren, verknüpfen, oder einfach über ein MIDI Keyboard spielen.
Mischen, Transponieren, Echoeffekte, Einbinden von Klängen in eigene Programme – alles kein Problem mehr.
Samplingsraten bis zu 36,4 KHz garantieren eine Qualität, die sich hören läßt.
Diese Leistungsmerkmale und vieles mehr bieten alle angebotenen Sampler.
Bereits die ST 1 / 88 schrieb über den AS SOUND SAMPLER II »Der Alleskönner«, die KEYBOARDS 12 / 87 schrieb »...verglichen mit 10 – 15 fach teureren Samplern... muß man von einer guten bis sehr guten Qualität sprechen«.

Der neue AS SOUND SAMPLER III setzt jedoch in Preis und Leistung völlig neue Maßstäbe. Erstmals können Sie einen 16 BIT Sampler mit CD Qualität unter 600 DM kaufen.

Umtauschaktion

(Gültig bis zum 31.6.88) Alle AS SOUND SAMPLER II Besitzer können diesen gegen den neuen 16 BIT AS SOUND SAMPLER III umtauschen.
Gegen Einsendung der AS SOUND SAMPLER II Hard- und Software, Kaufbeleg und Scheck über DM 305,— erhalten Sie den AS SOUND SAMPLER III.

1. AS SOUND SAMPLER III

16 BIT Sound Sampler mit allen Features des AS SOUND SAMPLERS II, in einer Qualität, deren Unterschied zur CD nicht mehr hörbar ist. Maximale Samplingfrequenz 44 KHz. Maximale Samplingzeit z.B. beim MEGA ST 4: ca. 160 sec.

Software und Hardware 16 BIT anschlußfertig für alle ATARI ST: **DM 598,—**
Soundbibliothek 10 Disketten MF 2 DD 16 BIT **DM 198,—**

2. AS SOUND SAMPLER II maxi

Durch den großen Erfolg der AS SOUND SAMPLER II Serie konnten wir den Preis erheblich senken, und bieten nun die bewährte Hard- und Software mit vielen Detailverbesserungen an.

Software und Hardware 8 BIT anschlußfertig für alle ATARI ST: **DM 298,—**
Soundbibliothek 10 Disketten MF 2DD 8 BIT **DM 149,—**

3. AS SOUND SAMPLER II Standard

Der AS SOUND SAMPLER II Standard bietet die gleichen Softwareleistungen wie der AS SOUND SAMPLER II maxi. Die Hardware wurde durch Einsparung des durchgezogenen Druckerports und des Peak Indicators, sowie ein schlichteres Design konsequent in Low Cost gebaut, jedoch ohne Verzicht auf die Samplequalität.

Software und Hardware 8 BIT anschlußfertig für alle ATARI ST: **DM 198,—**
Soundbibliothek 10 Disketten MF 2 DD 8 BIT **DM 149,—**

G DATA

Siemensstraße 16 4630 Bochum 1
Verkaufsbüro:
Hattinger Str. 312, 4630 Bochum 1
Telefonische Bestellungen:
023 25 / 6 08 97

Schweiz: Senn Computer AG
Langstr. 31
CH-8021 Zürich

Österreich: Computershop Rittner
Hauptstr. 34
A-7000 Eisenstadt

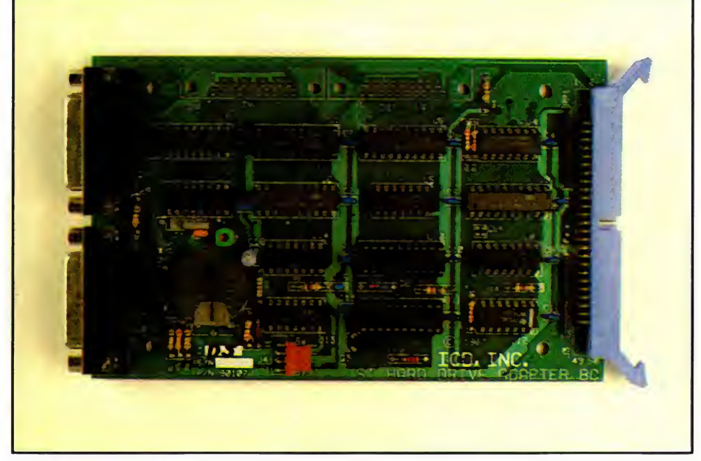
BENELUX:
G DATA BeNeLux
Postbus 70, NL-2000 AB
Haarlem
Tel. 023 / 32 13 31

Bitte 16 BIT!

Hiermit bestelle ich
___ Stk. AS SOUND SAMPLER II Standard zu DM 198,— / Stk.
___ Stk. AS SOUND SAMPLER II Maxi zu DM 298,— / Stk.
___ Stk. AS SOUND SAMPLER III 16 BIT zu DM 598,— / Stk.
☐ per Nachnahme (ca. 6,50 DM Versand)
☐ V-Scheck liegt bei (+ DM 5,—)
Absender:

Befehlssatz. Das liegt daran, daß kleine Befehlssätze in Hardware ausgeführt werden können, während große in sogenanntem Mikrocode implementiert werden können. Insgesamt ist ein Prozessor oft immer noch schneller, wenn er mehrere, extrem schnelle Befehle für eine Aufgabe braucht, als wenn er nur einen, langsamen ausführen muß. Der zweite Trick ist der Einbau eines schnellen Cache-Speichers. Wenn ein Prozessor immer alle Daten und Befehle aus dem großen, aber relativ langsamen Hauptspeicher holen muß, verliert er furchtbar viel Zeit. Deshalb implementiert man bei modernen Prozessoren einen Zwischenspeicher auf dem Chip, der erheblich schneller ist, als der normale Hauptspeicher. Das Besondere am Cache-Speicher des T800 ist seine Größe: 4 Kbyte. Darin kann man schon recht große Schleifen ohne jeden Zugriff auf den Hauptspeicher ausführen. Der T800 enthält eine Floating Point-Recheneinheit auf dem Chip. Der Arithmetik-Coprozessor ist also im Prozessor integriert. Die Floating-Point Recheneinheit des T800 ist extrem schnell. Sie leistet 1.5 Megaflop, also 1.5 Millionen Fließkommaoperationen pro Sekunde, wenn der Transputer mit 20 MHz getaktet wird. Eine 30 MHz-Version, die dann 2.25 Mflops leistet, ist in Vorbereitung. Der T800 rechnet schneller mit 32Bit Fließkommaals mit Integerzahlen. Insgesamt ist der Chip schneller als eine 68030/68882 Kombination. Der eigentliche Gag jedes Transputers sind aber seine vier seriellen Schnittstellen, die mit 20 Mbaud senden und empfangen können. Diese Schnittstellen ermöglichen es, mehrere Transputer zu Netzwerken zusammenzuschließen, die dann ein Vielfaches der Rechenleistung eines Einprozessorsystems erbringen. Das geht natürlich nur bei Problemen, die paralleles Rechnen ermöglichen. Glücklicherweise sind das, gerade in den Naturwissenschaften, ziemlich viele Probleme. Leider steckt aber auch die theoretische Erforschung von Parallelrechnern, und vor allem den dazugehörigen Betriebssystemen, fast noch in den Kinderschuhen.

Deshalb wäre die Idee der Schnittstellen alleine noch nicht besonders sinnvoll, erst mit einer speziellen Software-Unterstützung kann sich das Konzept von Prozessornetzwerken so richtig austoben. Deshalb hat die Firma Inmos, die den Transputer entwickelte, gleich eine spezielle parallele Programmiersprache entwickelt, die den Namen OCCAM, nach einem niederländischen Philosoph, trägt. Die Sprache wird oft als 'paralleles C' apostrophiert. OCCAM erlaubt die Programmierung von seriellen und parallelen Prozessen, die über Nachrichtenkanäle (in einem wirklichen System können das zum Beispiel die seriellen Schnittstellen, die den Namen 'Links', Bindeglieder tragen, sein) miteinander kommunizieren können. Dabei muß der Programmierer die Anzahl der im System vorhandenen Prozessoren nicht unbedingt kennen, das System kann, eine entsprechende Struktur der Transputerverschaltung vorausgesetzt, die Verteilung auf die vorhandenen Prozessoren selbst vornehmen.



Die Transputerkarte

Das ABAQ-System

Das Atari-Transputersystem enthält auf der Basisplatine einen T800-Transputer und 4 Megabyte Ram. Ein SCSI-Festplattenanschluß, der über einen DMA-Kanal arbeitet, ist vorhanden, eine 40 Mbyte-Platte soll im Grundsystem enthalten sein. Einer der vier Transputer-Links ist über ein Interface mit einem Mega-ST oder einer entsprechenden Platine im System selbst verbunden, die als Tastatur-, Maus- und Floppyinterface fungiert. Damit können alle ST-Peripheriegeräte angeschlossen werden. Die Bildschirmsteuerung übernimmt der Transputer auf der Hauptplatine. Für den Bildschirm steht 1 Megabyte Videospeicher zur Verfügung. Der Videospeicher besteht aus sogenannten Dual-Port-RAM, das sind Speicher, auf die zwei Teilnehmer gleichzeitig zugreifen können, der Prozessor und der Spezialblitter, der dem System zu enormen Grafikleistungen im 2D-Bereich verhilft. Bis zu 128 Millionen Pixel pro Sekunde können bewegt, Schrift kann mit bis zu 64 Millionen Pixeln pro Sekunde dargestellt werden; auch Farboperationen kann der Blitter ausführen.

Vier Grafikmodi kann das ATARI-System darstellen:

In der höchsten Auflösung stehen 1280*960 Pixel mit 16 Farben oder Graustufen zur Verfügung.

Die zweite Grafikstufe hat 1024*768 Pixel bei 256 Farben, die allerdings aus einer Palette von 16,7 Millionen Farben ausgewählt werden können.

Im dritten Grafikmodus gibt es zwei Bildschirme mit je 640*480 Pixeln und 256 Farben aus einer Palette von 16,7 Millionen.

Der letzte Modus ist die niedrig auflösende Grafik mit 512*480 Punkten, die eine gleichzeitige Darstellung aller 16,7 Millionen Farben erlaubt.

Auf der Hauptplatine sind drei Slots untergebracht, an denen der gesamte Systembus anliegt. Hier können Erweiterungskarten mit zusätzlichen Transputern, Peripheriekarten oder Speichererweiterungen eingesteckt werden. Auf eine Erweiterungskarte passen maximal 4 Transputer, so daß das System im Gehäuse auf maximal 13 Transputer mit einer Rechenleistung von maximal 20 Megaflop ausgebaut werden kann. Selbstverständlich können extern über die Transputerlinks beliebig viele Transputer angeschlossen werden. Auf der CeBIT wurde als Beispiel ein Fraktalbild mit 12 Transputern berechnet. Manche Systeme können so schnell nicht einmal Bilder laden.

Die Transputerlinks sind zwar schnell genug, um den Anspruch des Netzwerkanschlusses genüge zu tun, den eine Workstation nun einmal stellen muß, aber leider ist ein Transputerlink (noch) kein Standard. Deswegen sollte es nicht verwundern, wenn in Kürze eine Ethernet-Anschlußkarte für das System zur Verfügung stünde (Ethernet ist ein Standard-Netzwerk zur Rechnerkopplung).

Die gesamte Hard- und Software des ABAQ-Systems stammt übrigens nicht von ATARI, sondern von der Firma PeriHelion, die das System zusammen mit einer englischen Universität entwickelt hat.

Apropos Software,

die Software ist es schließlich erst, die ein System zum Leben erweckt. Dieser Gedanke führte bei PeriHelion zur Entwicklung des Betriebssystems Helios. Bisheriges Problem aller Transputersysteme war, daß kein geeignetes Betriebssystem, das die besonderen Features der Transputer ausreichend würdigte, zur Verfügung stand. Helios hat gute Chancen, ein Standard auf diesem Gebiet zu werden, denn es haben bereits andere Firmen Interesse an der Software bekundet. Meines Wissens hat bereits ein namhafter Hersteller von Transputersystemen Helios fest als Systemsoftware übernommen.

Helios ist ein Multitasking-Betriebssystem, das auf unterster Ebene weitgehend UNIX-kompa-

DAS GFA-BASIC



BUCH

KLAUS SCHNEIDER &
OLIVER STEINMEIER

Erste Tests haben gezeigt, daß das neue GFA-BASIC 3.0 die zur Zeit leistungsfähigste BASIC-Version auf dem ATARI ST ist. Der neue, extrem umfangreiche Befehlssatz erfordert eine grundlegend neue Einführung in die Programmiertechnik. Eine einfache Befehlsübersicht, wie sie mitgeliefert wird, genügt hier nicht.

Als optimale Ergänzung bietet sich dieses Buch an. In zwei Hauptteilen wird zunächst eine systematische Einführung in die Programmierung von BASIC unter Berücksichtigung der besonderen Fähigkeiten des neuen GFA-BASICs gegeben. Hier werden dem Neuling vom ersten Einzeiler bis zu abstrakten Datentypen alle Möglichkeiten der strukturierten Programmierung mit zahlreichen, durch Flußdiagramme transparent gemachten Beispielen nahegebracht. Doch auch BASIC-erfahrene Programmierer lernen hier die neuen Schleifenstrukturen (es gibt noch mehr als FOR, WHILE und REPEAT) kennen.

Der zweite Teil baut auf dem ersten auf und vermittelt weitere Kenntnisse der Programmierung



anhand von Programmen, die wiederum ausführlich beschrieben und erklärt sind. Hier seien eine Fakturierung, eine universelle Random-Access-Datenverwaltung sowie Beispiele zur Betriebssystemprogrammierung, insbesondere von Dialogboxen, genannt.

Durch zahlreiche Anhänge, die neben der obligatorischen ASCII-Tabelle auch einen Index, eine ausführliche Worterklärung sowie weitere nützliche Tabellen enthält, wird das Buch optimal ergänzt.

AUS DEM INHALT:

Erklärung der Schleifen- und Programmstrukturen
- Primzahlenberechnung
- Zahlenraten

Variablentypen und Arrays
- Sieb des Eratosthenes
- Adreßeingabe

Unterprogramme und Prozeduren
- Rekursionen
- Labyrinthsuche

Multitasking in GFA-BASIC

Abstrakte Datentypen
- Druckerspooler
- Verkettete Listen
- Binäre Bäume

Sequentielle Dateiverwaltung

Random-Access-Dateien
- Verkettete Listen auf der Diskette

Grafikprogrammierung
- Turtlegrafik
- Arbeiten mit mehreren Bildschirmen

Betriebssystemprogrammierung
- Aufrufen von TOS-Befehlen
- Verwenden des GEMs
- Menüverwaltung unter GFA-BASIC
- Arbeiten mit Dialogboxen

Beispielprogramme
- Fakturierung
- universelle Datenverwaltung
- etc.

Anhänge
- Worterklärungen
- Index
- sonstige Anhänge

ÜBER **450** Seiten
EINSCHLIESSLICH PROGRAMMDISKETTE

Dem Buch liegt eine Programm-Diskette bei mit sämtlichen Übungs- und Beispielprogrammen

BUCH & DISKETTE KOMPLETT
DM **59,–**

Preise sind unverbindlich empfohlene Verkaufspreise.

BESTELLCOUPON

Bitte senden Sie mir ________ St. DAS GFA-BASIC **3.0 BUCH** einschließlich Programm-Diskette für DM **59,–**
zzgl. DM 5,– Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnr. ____________________ PLZ/Ort ____________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

tibel ist. Erweitert ist das System aber um eine Prozessverwaltung für parallele Rechner, die es erlaubt, auch ohne Verwendung einer Spezial-Programmiersprache wie OCCAM Programme, für ein solches Multiprozessorsystem zu schreiben. Statt wie bisher Prozeduren mit Parameterübergaben zu programmieren, muß man die Modularisierung noch ein Stück weiter treiben. Parallel ablauffähige Programmteile müssen als unabhängige Prozesse geschrieben werden. Prozesse können untereinander mit Hilfe eines Nachrichtensystems kommunizieren. Die Verteilung der Prozesse im Rechnersystem übernimmt dabei das Betriebssytem, ebenso die Kommunikationsverwaltung. Der dabei verlorengehende Overhead, also die Rechenleistung, die bei den Verwaltungsvorgängen 'verlorengeht', ist dabei erstaunlich gering. Helios ist also ein sehr komfortabler Weg, parallele Programme zu entwickeln. An Systemsoftware werden zuerst C und Fortran, dann OCCAM, schließlich auch Pascal und Modula II zur Verfügung stehen, von dem unvermeidlichen Assembler mal abgesehen. Ein Debugger ist auch vorhanden.

Auf das Betriebssystem setzt die vom MIT entwickelte, netzwerkfähige Fensterverwaltung XWindows, die sich inzwischen als Standard etabliert hat (Vermutlich weil sie nicht nur leistungsfähig, sondern auch noch Public Domain ist) in der neuesten Version X11 auf. Der Vorteil bei der Verwendung dieses Fenster-Systems ist, daß bereits einige Anwendersoftware unter XWindows verfügbar ist, so daß relativ schnell benutzerfreundliche Software für das ATARI-Transputersystem entstehen könnte.

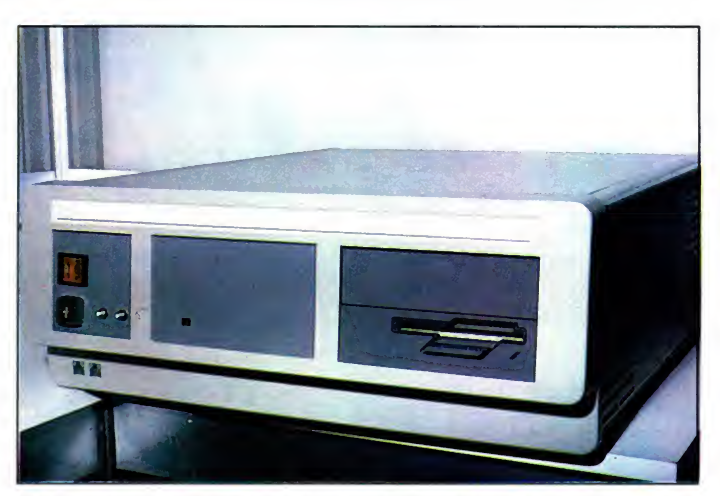
Der Konjunktiv hat seinen Grund: Im August sollen die ersten Maschinen ausgeliefert werden, wann aber das Betriebssystem endgültig fertig ist, wird sich noch zeigen. Versprochen ist es, zusammen mit den ersten Entwickler-Geräten für Ende April.

Das Atari-Transputersystem stellt in Sachen Rechenleistung völlig neue Maßstäbe in der Preisklasse unter 20.000 DM auf. Kein MacIntosh, kein IBM Modell 80, nicht einmal durchschnittliche Sun- oder Apollo-Workstations sind dem System in diesem Punkt gewachsen. Eine neue Technologie, die bisher kaum im Einsatz ist, hält hier gleich zu Beginn ihren Einzug im Low-Cost-Bereich. So etwas ist höchst ungewöhnlich. Ob es ein Erfolg wird, hängt im wesentlichen davon ab, ob Software-Entwickler das (Betriebs-) System akzeptieren. Man kann gute Hoffnung haben: Transputer-Software muß eigentlich immer so geschrieben sein, das sie auf nahezu jeder Hardware läuft, vorausgesetzt das Betreibssystem stimmt überein. Von Helios hängt also alles ab.

Eines ist jedenfalls ganz sicher: Mit dem Atari-Transputersystem sind die Zeiten, in denen die Preise einer Anwendungslösung im wesentlichen von der Hardware abhingen, endgültig vorbei. Mit Rechnern wie diesem wird der Anschaffungspreis für die Geräte nur noch einen Bruchteil der Softwarekosten für professionelle Lösungen, die schließlich auch professionellen Support verlangen, betragen. Mit den Transputern könnte ein neues Rechner-Zeitalter im Workstation-Bereich beginnen.

Der 68030...

Mit dem 68030 stellt Motorola den Nachfolger des 68020 vor, und damit in gewisser Weise auch einen Nachfahren des dem ATARI verwandten 68000. Die Neuerung bei diesem Chip liegt, neben der schon obligatorischen Geschwindigkeitssteigerung, in der auf dem Chip integrierten Speicherverwaltungseinheit. Entgegen vielen anderen Chipneuheiten benötigt der 68030 keine neue Software. Durch das aufwärtskompatible Codeformat wird es ermöglicht, bestehende 68000/68020-Software relativ problemlos zu übernehmen. Da ATARI ein auf dem 68030 basierendes System auf der CeBIT in Hannover vorgestellt hat, (zwar in der nichtöffentlichen 2. Etage, aber immerhin!), soll es hier schon einmal vorab angekündigt werden. Wann es überhaupt eine lauffähige Version für diesen Rechner geben wird ist zur Zeit noch nicht klar.



Der Prototyp des 68030-Rechners im vorläufigen Gehäuse.

Über diesen neuen Prozessor bzw. diesen neuen Rechner werden wir im nächsten Heft ausführlich berichten.

(MM/CB/CS/JW)

Es war einiges mehr zu sehen auf der CeBIT. Wie zum Beispiel Turbo C für den ATARI ST, oder die fertige Version von CALAMUS oder eine 100 MByte Festplatte, die von Weide vertrieben wird, oder....... aber wie schon oben erwähnt, wegen Platzproblemen ist es uns nicht gelungen, den gesamten Messebericht in dieser Ausgabe zu bewältigen. Deswegen haben wir uns für eine Teilung des Messeberichtes entschieden. In der nächsten Ausgabe werden wir über alle übrigen Neuigkeiten, die auf der Messe zu sehen waren berichten.



ENDE

VOM PAPIER ZUR DATENBANK

Die elementaren Datenbankbefehle

Hallo! Ich begrüße Sie zum zweiten Teil des Programmierkurses. Die heutige Folge beschäftigt sich mit den wichtigsten Befehlen zum Arbeiten mit Relationen. Ich erwähnte bereits in der Einleitung, daß ich nur auf zwei Systeme eingehe, dBMAN und STandard BASE. Dies hat folgende Gründe:
Beide Programme orientieren sich stark (STBASE sogar 100%) nach dem Standard-Datenbanksystem dBASE III für PC's. Während meiner Ausbildungszeit (DV-Kaufmann) habe ich damit intensiv gearbeitet und mich sehr dafür begeistert.

ADIMENS hingegen ist zwar auch eine relationale Datenbank, die aber in der Programmierung und Bedienung erheblich von den oben genannten abweicht. Außerdem meine ich, ist sie nicht so flexibel gestaltbar und eher für Endbenutzer als für Programmierer konzipiert. Mit der Sprache ADITALK (siehe Test ST-Computer 3/88) erhielt ADIMENS eine 'Query-Language' (Abfragesprache) sowie die Möglichkeit, den Verarbeitungsablauf über Schleifen und Bedingungen zu steuern. Vielleicht werde ich für spätere Folgen bei entsprechendem Interesse und Resonanz dieses Programm mit in den Kursus einbeziehen.
Soviel dazu und zum Thema: ich beschreibe die Befehle, um mit Hilfe des Datenbanksystems (DBMS = DATA BASE MANAGEMENT SYSTEM) Dateien aufzubauen.
Ich werde sie verständlicherweise nicht in allen Einzelheiten ausführen, denn ich will ja keine Kopie der Handbücher schreiben. Mein Kurs soll Ihnen den Einstieg in die relationale Welt Schritt für Schritt erleichtern. Um alle Fähigkeiten des Systems auszunutzen, müssen Sie auf die vollständige Befehls-Syntax in den Handbüchern zurückgreifen.
Der nun folgende Teil ist in fünf Abschnitte gegliedert:

a) die Umsetzung der Relationen in eine Struktur (Aufbau einer Tabelle), sowie deren ggfs. später notwendige Modifikation
b) Erfassen von Daten,
c) Abrufen (Anzeigen und Suchen) von Informationen,
d) Änderungen vornehmen,
e) Daten löschen

Startet das Datenbankprogramm mit einem Doppelclick auf dem entsprechenden Symbol. Es meldet sich dann der Prompt (bei dBMAN in der obersten Zeile CMD:, STBASE richtet sich nach dem großen Vorbild).

Vorsicht!

Das Programm muß IMMER mit einem QUIT verlassen werden. Nur so ist sichergestellt, daß alle Daten, die sich noch im Rechner befinden, auf Disk zurückgeschrieben werden. Falls Sie einfach einen Reset machen, um ins Desktop zurückzugelangen, könnte ein völliger Datenverlust die Folge sein, weil die Dateien nicht sauber geschlossen werden. Das gilt auch für bereits bestehende Files, weil u.U. der Datei-Ende-Pointer (EOF) nicht richtig gesetzt wird.
Beide Systeme erlauben drei Betriebsarten: zum einen den Kommando-Modus, in dem man sich nach dem Start des Programms befindet (erkennbar am Prompt). Alle eingegebenen Befehle werden sofort interpretiert, ausgeführt und Ergebnisse am Bildschirm protokolliert.
Im Programmier-Modus steht ein Full-Screen Editor zur Verfügung. Es können richtige Programme geschrieben werden. Doch dazu im übernächsten Teil des Kurses mehr.
In der dritten Betriebsart, vergleichbar mit 'RUN' beim BASIC, übernimmt eine erstellte Programmlogik die Steuerung des Verarbeitungsablaufs.

Schaffe, schaffe, Häusle baue...

Hier sollen natürlich keine Häuser, sondern Tabellen geschaffen (kreiert) werden. Abbildung 1 zeigt eine Relation, die in eine dem Computer verständliche Form umgesetzt werden soll. Ich werde mich auf dieses Beispiel, welches nur als Testobjekt dient, des öfteren beziehen. Der Befehl, mit dem Relationen in Ihrer Struktur definiert werden, lautet: **CREATE name <RETURN>** (Anmerkung: ich schreibe aus dem Grund der Übersichtlichkeit alle Schlüsselwörter in Großbuchstaben; es ist aber nicht nötig, denn das System erkennt auch Kleinbuchstaben. STBASE bietet die Möglichkeit, alle Schlüsselwörter mit ihrer Kurzschreibweise zu verwenden, d.h. nur die ersten vier Buchstaben reichen aus).
Mit (name) gibt man der Tabelle einen Namen. Unter dieser Bezeichnung wird die Datei auf der Disk gespeichert und unterliegt so den

name	plz	ort	alter	datum	groesse
Meier	2000	Hamburg		01.01.88	1.52
Müller	8000	München		02.01.78	1.78
Otto	6000	Stuttgart		04.01.66	1.65
Fuchs	1000	Berlin		06.12.67	1.55
Fehrmann	2000	Hamburg 100		25.05.45	1.92
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.

Abb. 1: Beispieltabelle

TOS-Konventionen (Buchstaben und Ziffern, max. 8-stellig). Die Extension darf nicht mit angegeben werden; sie lautet immer .dbf (data base file, in Anlehnung an dBASE III).
Der Tabellenname muß einmalig vergeben werden und darf nicht bei einer weiteren Relation des gleichen Anwendungsgebietes verwendet werden, weil sonst die alte überschrieben wird (Beispiel für gültige Namen: DISK, PROGRAMM, TEST1). Das erscheint logisch, da in einem Diskverzeichnis nicht zweimal der gleiche File-Name auftreten kann.
Das DBMS stellt daraufhin eine Erfassungsmaske auf dem Bildschirm zur Verfügung, die sich in vier Spalten aufteilt. Mit der <RETURN>-Taste bewegt man sich zur jeweils nächsten Kolumne.

a) Feldname: Jede Domäne einer Tabelle muß einen Feldnamen erhalten, der in einer Relation einmalig sein muß; in einer anderen kann er aber erneut vergeben werden. Die Namenskonvention von Feldern: bis zu 10 Zeichen; erstes Zeichen muß ein Buchstabe sein, danach können Ziffern folgen, auch der Unterstrich (_) ist erlaubt (Beispiel: DISK_NR, ANREDE). Mit diesem Namen erfolgt der Zugriff auf die Feldinhalte.

b) Feldtyp: Er bestimmt, welche Art von Informationen die Felder dieser Spalte beinhalten werden.

Das System bietet eine Vielzahl von Möglichkeiten an.

C- alphanumerisches Feld (z.B. für Namen, Anschriften)

N- numerisches Feld, mit dem gerechnet werden soll

D- Datumsfeld. Das System enthält spezielle Kalenderfunktionen, mit denen Datumsangaben behandelt werden (Prüfung auf Gültigkeit, Vergleiche, usw.), z.B. wird bei der Eingabe der 29.02.87 abgelehnt, weil es ein ungültiges Datum ist.

Es gibt noch weitere Typen, die aber selten benutzt werden bzw. durch andere ersetzbar sind (z.B. logische Felder) und deshalb hier nicht beschrieben werden sollen. Die genannten vier Typen decken fast alle Aufgabengebiete ab.

c) Länge: sie gibt die Breite des Feldes an. Es muß die Gesamtlänge eingegeben werden, z.B. muß bei einer dreistelligen Zahl mit zwei Nachkommastellen hier eine sechs (6) eingetragen werden.
Bei Datumsfeldern (Typ D) ist hier keine Angabe nötig; die Länge wird automatisch nach Drücken von <RETURN> auf 8 vorgegeben.

d) Nachkommastellen: Für numerische Felder kann optional eine Stellenzahl nach dem Komma erfaßt werden. Diese kommt in das vierte Feld der Erfassungsmaske.
Sind alle Felder beschrieben, beendet man mit CTRL-S den Aufbau. Die Datei, deren Strukturdefinition in Abbildung (2) wiedergegeben ist, steht nun zur Verfügung.
Mit dem Befehl **USE name** eröffnet man die Tabelle zur Bearbeitung. Falls es erforderlich wird, sich später diese festgelegte Struktur der Datei anzuschauen, so kann der Aufbau mit **LIST STRUCTURE** oder **LIST STRU (STBASE)** angezeigt werden. Stellt man fest, daß die Tabelle in ihrer Struktur geändert werden muß (z.B. Feldlänge zu klein, Hinzufügen weiterer Felder, Löschen von Feldern, Umbennung von Feldnamen usw.) ermöglicht **MODIFY STRUCTURE** oder **MODI STRU** die nachträgliche Änderung des Dateiaufbaus. Dabei gehen bereits erfaßte Informationen nicht verloren, es sei denn, man entfernt oder verkleinert Felder.
Das DBMS legt zur Strukturänderung eine Kopie der Datei an, ändert das Original und kopiert dann aus der Zwischendatei die Daten zurück. Die Änderungen können in der bereits bekannten Maske vorgenommen werden.

Das Erfassen von Daten

Für die Benutzer von dBMAN gilt: Das Datumsformat ist grundsätzlich in der Schreibweise MM/TT/JJ. Da dies in Deutschland nicht üblich ist, muß in die Kommando-Zeile **SET DATE TO 1** eingegeben werden, um die "normale" Form (TT/MM/JJ) zu erhalten. Besonders bei Eingaben ist dieser Punkt wichtig, weil das System sonst beharrlich z.B. den 25.01.88 ablehnt (es interpretiert daraus nämlich einen 25. Monat, den es ja nicht gibt).
Bei STBASE braucht keine Angabe gemacht zu werden (das System 'schluckt' zwar den Befehl SET DATE GERMAN; er ist aber ohne Wirkung). Die Datumseingabe geschieht in deutschem Format; leider wird es umgekehrt abgespeichert (MM/TT/JJ). Zum Aufnehmen neuer Informationen stehen zwei Befehle bereit. Zum einen kann man an beliebigen Stellen Datensätze einfügen (INSERT, nur bei STBASE) oder an das Ende neue Tupel anfügen (APPEND). Da unsere Beispieltabelle noch leer ist, benutzen wir den APPEND-Befehl und schreiben einfach in die Kommandozeile **APPEND** und erfassen so den ersten Datensatz der Tabelle mit Hilfe einer Ediermaske. Links befinden sich die Datenfeldbezeichnungen, rechts die Eingabefelder.

```
name       plz    ort              alter    datum     groesse
-----------------------------------------------------------------
Meier      2000   Hamburg                   01.01.88  1.52
Müller     8000   München                   02.01.78  1.78
Otto       6000   Stuttgart                 04.01.66  1.65
Fuchs      1000   Berlin                    06.12.67  1.55
Fehrmann   2000   Hamburg 100               25.05.45  1.92
  .          .      .                          .        .
  .          .      .                          .        .
  .          .      .                          .        .

  .CREATE datei  (STBASE).  CMD:CREATE datei  (dBMAN)

  Feldname      typ    länge    dez.stellen
  --------------------------------------------
  name          c      20       0
  plz           n       4       0
  ort           c      20       0
  alter         n       2       0
  datum         d       8 (wird automatisch eingefügt)
  groesse       n       4       2
```

Abb. 2: Festlegen einer Struktur

Man merkt, daß in numerisch definierte Felder nur Ziffern eingegeben werden können. Bei Dezimalzahlen erscheint stellengerecht das Komma (als Punkt). Das System überprüft automatisch die eingegebenen Daten auf Gültigkeit und erspart so dem Programmierer die umständliche Codierung von Plausiblitätskontrollen. Sogar Datumsangaben werden kontrolliert und logisch falsche Eingaben abgewiesen.

Wir füllen, um Testdaten für folgende Übungen vorzubereiten, die Datei mit ca. 30 Phantasiedaten. Lassen Sie bitte das Feld 'alter' noch leer; es wird zur Verdeutlichung eines anderen Befehls im 'jungfräulichen' Zustand benötigt.

Innerhalb der Erfassungsmaske können die Cursor-Tasten zum Edieren benutzt werden. Nach der Eingabe des letzten Feldes eines Satzes erscheint sofort eine neue, leere Maske (bei dBMAN muß CTRL-N gedrückt werden).

Sollen keine weiteren Daten erfaßt werden, so drückt man nach Erfassung des letzten Satzes CTRL-S. Es meldet sich dann der Prompt des Systems wieder.

Zum Verständnis: das System führt intern einen Zeiger, der auf den aktuellen Datensatz verweist (Datensatznummer). Will man zwischen den ersten und zweiten Satz einen neuen einfügen, so schreibt man **GO 1** und anschließend **INSERT.** Mit dem GO-Befehl setzt man den Datensatzzeiger auf einen ganz bestimmten Satz. Jeder Satz ist so einzeln ansprechbar. GO TOP setzt den Zeiger auf den ersten, GO BOTTOM auf den letzten Satz der Tabelle. Mit **SKIP n** bewegt man sich relativ zur aktuellen Position. Ist n positiv, so springt man Richtung Datei-Ende, sonst zum Anfang. So gelangt man mit SKIP 1 zum nächsten Satz (hier könnte die '1' weggelassen werden); mit SKIP -2 setzt man den Zeiger auf den vorvorletzten Datensatz. In der internen Variablen RECNO() ist dieser Zeiger gespeichert. Anzeigen mit: **? RECNO()** (Das Fragezeichen ist ja wohl vom BASIC her als PRINT - Befehl hinlänglich bekannt !!) Mit dem INSERT werden die Datensatznummern der folgenden Zeilen jeweils um 1 erhöht. Der Befehl kann auch mit dem Zusatz BEFORE versehen sein. Dann erfolgt die Einfügung vor der aktuellen Zeile.

Das Abrufen von Informationen

Am LIST bzw. LOCATE-Befehl erkennt man deutlich den descriptiven Charakter der Sprache. Die Formulierung kommt der Umgangssprache sehr nahe. Mit **GO TOP** und anschließend **LIST ALL** werden alle Datensätze gelistet (ei, wer hätte das gedacht ?). Ganz links vor jeder Zeile steht die Nummer, unter der der Satz gespeichert ist. Sie kann mit der Erweiterung **LIST OFF ALL** unterdrückt werden.

Sind in der Tabelle viele Sätze gespeichert, so scrollen sie schnell über den Bildschirm. Der DISPLAY-Befehl verhindert das Scrollen und stoppt die Ausgabe nach einer Seite und wartet auf einen Tastendruck zum Weitermachen (**DISPLAY ALL**). ALL gibt den Bereich an, der gezeigt werden soll. Möglich wäre auch ein **LIST NEXT 5**, welches die nächsten fünf Sätze ab der aktuellen Position listet, sofern vorhanden.

Dies ist aber nur die einfachste Form der Anwendung. Man kann Bedingungen stellen und so einzelne Sätze selektiv herausgreifen. Möchte man alle Personen des Wohnortes Hamburg, so schreibt man **LIST ALL FOR ort = 'Hamburg'** zum Anzeigen ausgewählter Datensätze. (<ort> ist der Feldname !) Als Vergleichsoperatoren sind erlaubt: < > =, sowie deren Kombinationen (<>, >=, <=). Die Operationen sind für alle vier genannten Feldtypen erlaubt, sogar bei Datumsfeldern: **LIST ALL FOR datum > CTOD ('02.01.77')** Die Funktion CTOD bedeutet 'character to date' und wandelt ein alphanumerisches Feld in den Typ 'Datum' um. Der Inhalt bleibt derselbe. Andernfalls könnte die Operation nicht ausgeführt werden, da stets auf beiden Seiten die Datentypen übereinstimmen müssen. Ein numerisches Feld kann z.B. auch nicht mit 'ABC' verknüpft werden. Das DBMS stellt aber für alle Fälle die nötigen Umwandlungsfunktionen bereit:

STR(alter,n) wandelt numerisch in alphanumerisch mit der Feldlänge n.

DTOC(datum) Datum nach Character

VAL(name) konvertiert String in Zahlen, beginnend von links. Beim Auftreten von Buchstaben bricht die Funktion ab, Beispiel: VAL('12AB') ergibt 12

Leider (s.o.) sind bei STBASE die Datumsfelder im Monat und Tag vertauscht gespeichert. So wird aus dem 02.01.77 der 1. Februar und nicht der 2. Januar. Darauf müsen Sie achten !! Es ist möglich, auch auf Teile eines Feldes zuzugreifen. Der Substring-Operator $ ermöglicht die Suche eines Teilstrings innerhalb eines bestimmten Feldes.

LIST ALL FOR 'burg'$ort Diese Anweisung sucht alle Orte, in denen 'burg' vorkommt. Dabei ist es unerheblich, an welcher Stelle sich diese

Kombination befindet. Man findet also 'Hamburg' wie auch 'Hamburg 100'. Die Ausgabe über Drucker ist mit dem Zusatz 'TO PRINT' möglich.
Sollen die Datensätze nicht alle nacheinander, sondern einzeln ausgegeben werden, so bietet sich der Befehl LOCATE an. Er ist in der Wirkungsweise ähnlich wie LIST, bleibt jedoch nach dem ersten zutreffenden Datensatz stehen, der dann mit DISPLAY angezeigt werden kann. Mit CONTINUE wird die Suche fortgesetzt.
LOCATE ALL FOR 'A'$name Ein folgender DISPLAY zeigt den gefundenen Datensatz an. Einzelne Felder sind über ihren Feldnamen abrufbar:
? ort oder **? DTOC(datum)** (Datumsfelder können ohne Umwandlung s.o. nicht angezeigt werden !!)

Änderungen

Daten unterliegen ständigen Änderungen, sei es, daß Falscheingaben korrigiert oder alte Daten an aktuelle Werte angepaßt werden müssen.
Die Datenbanksysteme bieten mit dem Befehl **EDIT** bzw. **REPLACE** die Möglichkeit, bestehende Daten mit neuen Werten zu überschreiben. Dazu muß der Datensatzzeiger auf den zu ändernden Satz gestellt werden, z.B. GO 5. Man kann aber auch einfacher schreiben **EDIT 5** Damit wird die Maske (wie bei APPEND) angezeigt, in der die alten Daten modifizierbar sind.
Mit dem REPLACE-Befehl können einzelne Felder sofort geändert werden, ohne die Maske anzuzeigen. Dies ist besonders später in Programmen nötig.
So erfolgt mit **GO 5** und anschließendem **REPLACE name WITH 'Egon Müller'** die Änderung des Namens im fünften Satz. Wie bei allen anderen Befehlen kann die Verarbeitung auf mehrere Sätze gleichzeitig wirken:
REPLACE ALL alter WITH YEAR(DATE())-YEAR(datum)
Whow, das ist ein Ausdruck! Er ersetzt alle Felder, in denen das Alter steht (zunächst noch leer, s.o.) mit dem errechneten aktuellen Alter. Und das geht so:

- **DATE()** enthält das Systemdatum
- die Funktion **YEAR(...)** ermittelt das Jahr aus einem Datum

Vom heutigen Jahr wird also das Geburtsjahr abgezogen und in das Feld 'alter' gestellt. Dies geschieht bei allen Datensätzen !! Mit LIST ALL kann anschließend das Ergebnis kontrolliert werden.

Löschen (Ja/Nein)???

Soll einzelner Satz aus einer Tabelle entfernt werden, benutzt man **DELETE**, womit der aktuelle Satz gelöscht wird - nein, noch nicht endgültig. Er kann mit RECALL wieder zum Leben erweckt werden. Soll die Löschung aber tatsächlich durchgeführt werden, so muß nach dem DELETE ein PACK folgen, damit die fortlaufende Datensatznumerierung wieder in eine kontinuierlich aufsteigende Reihenfolge gebracht wird. Löschen wir nämlich aus einer Tabelle mit 5 Sätzen den ersten, so müssen die übrigen eine neue Nummer erhalten (1...4).
Der 'DELETE' markiert lediglich die Datensätze mit einem Zeichen (*), daß sie zum Löschen vorbereitet sind. Um diese Sätze von der weiteren Verarbeitung auszuschließen, formuliert man **SET DELETED ON**. Dann werden mit LIST ALL die markierten Sätze nicht mehr mit ausgedruckt. SET DELETED OFF behandelt die markierten Zeilen, als wären sie nicht gelöscht.
Um alle Sätze einer Relation zu löschen, könnte man z.B. so vorgehen **DELETE ALL** und **PACK**. Dieser Vorgang würde aber recht lange dauern, da erst jeder Satz zum Löschen markiert und anschließend physisch auf der Disk entfernt werden müßte. Eleganter und sehr viel schneller arbeit der ZAP. Er löscht in einem Arbeitsgang (nach einer Sicherheitsabfrage) alle Sätze einer Datei, nicht jedoch die Struktur!
So, das waren die grundsätzlichen Befehle, um mit einer Relation zu arbeiten. Es dürfte Ihnen nun nicht mehr schwerfallen, die Tabellen des ersten Teils zu codieren (mit CREATE) und anschließend mit Daten zu füllen. Probieren Sie es doch einfach aus. Beachten Sie aber bitte:
Bevor Sie die Arbeit mit dem Datenbanksystem beenden, vergessen Sie das KWITT nicht!!!!!

Schwester, bitte Skapell !!

Ich sprach im ersten Teil von Mengen- und Relationenoperationen, auf die ich noch kurz eingehen möchte.
a) Mengenoperationen beziehen sich immer auf die gesamte Tabelle. Als Beispiel nenne ich den SUM-Befehl, der die Summe aller Feldinhalte einer numerischen Domäne ermittelt (hier: Berechnung des Gesamtalters aller Personen). Das Ergebnis wird auf dem Bildschirm angezeigt.
SUM alter - Die Ausführung dieses Befehls kann von Bedingungen (FOR) anhängig gemacht werden (siehe LIST).
Ein weiterer Befehl, der die Tabelle als Einheit betrachtet, ist **COUNT**. Er zählt alle Datensätze; auch in Verbindung mit FOR, z.B.
COUNT FOR name = 'Meier' ermittelt, wieviele Personen unter diesem Namen erfaßt sind.
Ein 'ganz toller' Befehl ist SORT. Mit dessen Hilfe kann eine gesamte Datei nach bestimmten Feldern auf- oder absteigend sortiert werden. Als Standardwert gilt eine aufsteigende Sortierfolge:
SORT ON name TO temp Einfach, oder ? Das klingt doch echt wie normale Sprache. Programmieren Sie so etwas mal in BASIC oder C !!!
Die nach Namen sortierte Datei ist in der Tabelle temp gespeichert. Mit **USE temp** und **LIST ALL** können wir uns das Ergebnis betrachten.
Leider erzeugt ein SORT eine sortierte Kopie der alten Datei. Dies kann bei großen Datenmengen zu Engpässen führen (besonders auf Disketten). Auch ist die Tabelle immer nur unmittelbar nach dem SORT in richtiger Reihenfolge. Nachträgliche Änderungen erfordern grundsätzlich eine Neusortierung.

In der nächsten Ausgabe werde ich eine elegantere Form der Sortierung beschreiben, das Indizieren. Hierbei werden neu eingegebene Sätze sofort in die richtige Reihenfolge gebracht.
b) Relationenoperationen ermöglichen die gezielte Auswahl von Zeilen und Spalten. Die Sicht auf eine Tabelle wird eingeschränkt; die Datei erscheint kleiner, obwohl mehr Sätze bzw. Felder gespeichert sind.
LIST name,ort zeigt aus der Relation nur die angegebenen Felder. Die anderen Spalten sind nicht erreichbar.
SET FILTER TO plz = 8000 und **LIST ALL** (Der Vegleichsoperator darf hier nicht in Hochkommata eingeschlossen sein, weil es sich um ein numerisches Feld handelt!) Mit dieser Anweisung werden nur noch die Personen aus München betrachtet. Die Tabelle ist scheinbar geschrumpft, denn selbst mit LIST ALL werden nicht alle gespeicherten Sätze gelistet. Der Filter blendet also Zeilen aus und kann auch auf mehrere Spalten gleichzeitg wirken (mit einer logischen UND-Verknüpfung)
SET FILTER TO plz = 8000 .and. datum = CTOD('01.01.77')
(Merke: das Setzen eines neuen Filters hebt die vorherige Wirkung wieder auf) Wenn ich schon dabei bin: Folgende logische Operationen sind möglich:
.NOT. nicht **.AND.** und **.OR.** oder
Sie können nicht nur beim Filtern, sondern auch beim LIST, LOCATE und DELETE verwendet werden, um die Verarbeitung von Bedingungen abhängig zu machen.
Ein **SET FILTER TO** ohne Parameter stellt den ursprünglichen normalen Zustand her. Übrigens, wenn Sie die dauernden Systemmeldungen stören, dann stellen Sie sie doch einfach ab. Dann werden aber auch Ergebnisse (z.B. COUNT, SUM) nicht mehr angezeigt.
SET TALK OFF Das will ich jetzt auch tun und beende damit den zweiten Teil. Hoffentlich habt Ihr anhand ausgesuchter Beispiele die Leistungsfähigkeit des DBMS erkannt. Gerade bei der Verarbeitung von riesigen Datenmengen macht sich der Einsatz in einer enormen Arbeitserleichterung bemerkbar.
Freut Sie sich schon auf die nächste Ausgabe. Denn da geht ganz fix die Post ab mit Sortierung und Datensuche.

Paul Fischer

Quellen: dBASE III Das relationale Datenbanksystem für 16-Bit- Computer.
Autor: Dr. Peter Albrecht,
Verlag: Markt und Technik

ENDE

Mini Clip

Der ATARI ST in der Praxis

Unter diesem Motto möchten wir eine Reihe von praktischen Anwendungen auf dem ATARI ST vorstellen. Den Anfang machte jetzt der Artikel 'JuriSTische Anwendungen'. Damit dies nicht der letzte seiner Art bleibt, möchten wir hiermit Sie, liebe Leser, dazu aufrufen, uns Ihre Erfahrungen und Anwendungen mit dem ATARI ST im alltäglichen Leben mitzuteilen.

Vielleicht machen Sie Ihre Buchführung, verwalten Ihre Kunden oder steuern Ihre Meßgeräte mit dem ST. Alles, was Sie für einen Leserkreis, wie den der ST-Computer, interessant halten, können Sie uns mitteilen. Im Zweifelsfalle rufen Sie einfach die Redaktion unter der aufgeführten Adresse an.

Veröffentlichte Erfahrungsberichte werden natürlich angemessen honoriert.

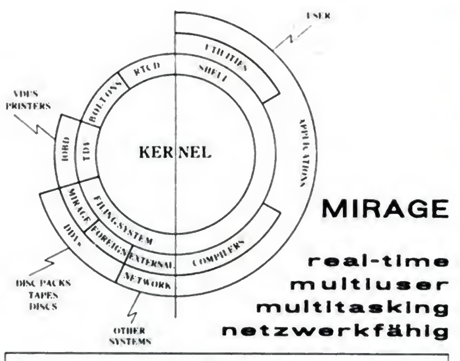
Bitte beachten Sie bei der Einsendung:

Schicken Sie den Bericht und eventuelle Hardcopies auf Papier und Diskette. Die Diskette wird zurückerstattet.

Einsendungen direkt an die Redaktion:

Merlin-Computer GmbH
ST-Computer Redaktion
'Anwendungen'

Industriestr. 26
D-6236 Eschborn
Tel.: 06196/481811



Die Alternative zu UNIX. Seit Jahren hat **MIRAGE** sich auf dem Markt der Workstations fest etabliert. Jetzt gibt es die ST-Version. Extrem schnell. Extrem vielseitig. Einige Features:

- Lesen & Schreiben von TOS-Disketten
- TOS-und **MIRAGE**-Partitions auf **einer** Festplatte
- TOS-Calls von **MIRAGE** aus möglich (**kein** GEM)
- Line-A Grafik zugänglich
- virtuelle Bildschirme (umschaltbar per Tastatur)
- bis zu 128 dynamische Tasks pro Job
- Kommandosprache mit Variablen, Nesting etc.
- geräteunabhängiges Treiberkonzept
- schneller, komfortabler Editor
- Supervolumes: Eine Datei über mehrere Drives
- echtes Netzwerk ohne zusätzliche Hardware
- Sprachen: Basic, Pascal, Lisp, Fortran, C, APL

Der **Assembler**:

- Macros (mit lokalen Labels und Konstanten!)
- bedingte Assemblierung
- Unterstützung von 68020/68030/68881/68882
- Fließkomma-Konstanten
- Linker, Debugger, Profiler, Library-Funktionen

MIRAGE: Die ideale Lösung für Software-Entwickler, die eine preiswerte Workstation-Umgebung brauchen.
MIRAGE: Für Unis, die z.B. ihre Fortran-Anwendungen aus dem Rechenzentrum lösen wollen (Swifte-Fortran -77 ist als **Error-Free** zertifiziert, die **NAG**-Bibliotheken sind erhältlich - auch auf Disketten!).
MIRAGE: Für Meßdatenerfassung & Steuerzwecke.
MIRAGE: Für die Portierung von PC-Software auf den ST (z.B. mit Lattice C); für Netzwerke; ...

MIRAGE-ST (ROM-Pack, Disks, Manuals) **348.-**
MicroGrid (Netztreiber für RS232 u. MIDI) **98.-**
- Bitte fordern Sie Infos und Preislisten an -

gdat Stapelbrede 39 4800 Bielefeld 1 **0521/875 888**

We don't need no education



We don't need no thought control

APL/68000 - Das ganz besonders schnelle APL für 68000er Rechner. **Von MicroAPL.** Die Sprache der Profis für ATARI, MAC und AMIGA - oder darf es eine Workstation sein? Bei **gdat** zum supergünstigen Preis: Incl. Support, Update-Service und Toolsdiskette (nur ST). **Wir liefern immer die neueste Version.** Literatur- und Referenzhinweise sowie ausführliche Infos gratis.

APL/68000-ST+ **298.-**
Die PLUS-Version wird exklusiv von **gdat** vertrieben und enthält wertvolle Erweiterungen und Tools, darunter sogar eine kleine Datenbank!

APL/68000 Language Manual **49.-**
Der Kaufpreis wird bei nachfolgendem Kauf des APL voll angerechnet.

APLPRINT **198.-**
druckt Text und Grafik unter APL.
LineAl **164.-**
Schnelle Line-A Grafik.
APL-ASS **164.-**
Für Assembler-Routinen in APL.
APL-EDIT **248.-**
Variablen-Editor, Icons, Fonts ...
GD_GRAPH **48.-**
VDI-Grafiktools.

APL ist eine Hochsprache der vierten Generation, die seit Jahren ihre Effizienz erwiesen hat. Und: APL ist eine Sprache zum Anfassen - Leicht zu lernen, intuitiv, fehlertolerant. Mit APL werden Probleme gelöst, nicht geschaffen. Zudem können wir uns als zuverlässigen Partner empfehlen: **gdat** arbeitet seit 7 Jahren mit APL/68000. Wir wissen, wovon wir reden. Fordern Sie das Info an.

gdat Stapelbrede 39 4800 Bielefeld 1 **0521/875 888**



Replica Box ST

– Copy-Station –

Hardwarezusatz, der es Ihnen ermöglicht, von jedem Ihrer Original-Programme eine Arbeitskopie anzufertigen.

Die Replica-Box wird zwischen Computer und Diskettenstation gesteckt und übernimmt die volle Kontrolle über das Laufwerk.

Lieferumfang: Replica-Box mit Gehäuse, Software, Beschreibung auf Diskette.

Preis: **249,– DM**

Exclusiv bei:

Hendrik Haase Computersysteme
Wiedfeldtstr. 77 · D-4300 Essen 1
Tel. 02 01 / 42 25 75

KaroSoft
ST - Soft-u. Hardware Vertrieb
Biesenstraße 75 4010 Hilden Telefon 02103/42022

Atari ST

Anwenderprogramme:

ADIMENS Datenbank	189,–
ISGEMDA Datenbk. System dt.	229,–
STEVE V. 3.0	448,–
CopySTar V. 2.2	142,–
Timeworks DTP (GST)	349,–
CALAMUS DTP (DMC)	849,–
Signum II Text/Grafikprgr.	369,–
STAD	159,–
Flexdisk	66,–
1st Proportional	85,–
Printmaster Plus	95,–
BS – Handel	498,–
BS – Fibu	598,–
BS – Timeadress	149,–
BSS – Plus, Softw. Baukastensystem f. Handwerk u. Handel, Info anfordern	
MEGAMAX-MODULA-2, kpl. in dt.	368,–
STAR-WRITER-ST	189,–
GFA-Farbkonverter	59,–
GEM-Retrace – Recorder	92,–
Sympatic – Paint (G DATA)	268,–
PC-ditto MS-DOS Emulator	198,–
T.i.M. Buchführung	269,–
SALIX-Prolog Comp./Interpr.	195,–
GFA-BASIC Interpr./Comp.	je 88,–
monoSTar plus	139,–
EXERCISE, Englischlernprgr.	85,–
Pro Sound Designer, neue V.	169,–
fibu MAN m, Fibu m. BWA	938,–
fibu MAN f, Finanzbuchh.	738,–
fibu MAN e, Einn./Übersch.	368,–
G Copy	89,–
G RAMdisk II	42,–
Interprint II m. RAMdisk	89,–
Interprint o. RAMdisk	44,–
Harddisk Help u. Extension	119,–
G Diskmonitor II	89,–
G Datei	179,–

Spiele:

ASTERIX, dt.	54,90
Vermeer, dt.	69,–
Karting Grand Prix	29,90
Jagd auf roter Oktober, dt.	67,50
500 ccm Grand Prix Sim.	59,50
Universal Military Sim., dt.	72,50
Chamonix Challenge, dt.	59,90
Enduro Racer, Sim. dt.	59,90
Super Star Eishockey, dt.	69,–
STAR TREK, dt.	59,90
Test Drive, dt.	79,–
Dungeon Master, dt.	69,–
Bubble Bobble	59,–
Kaiser	119,–
Flight Simulator II sw/col.	119,–
Flight Sim. Scenery Disk	59,–
Crazy Cars	59,–
Western Games, dt.	55,90
Black Lamp, dt.	59,–
Bad Cat, dt.	49,90
Defender of the Crown	67,90
Leader Board Golf	69,90
Leader Board Tournament	35,–
Gunship, Helicopter Sim.	72,50

Hardware:

Mausmatte	17,9
Pro Draw Grafiktablett	875,–
Scanner „Hawk" CP 14 ST	2750,–
A-Magic Turbo-Dizer	265,–
Weide 3,5" Laufwerk 1 MB	378,–
Weide 3.5" Laufwerk 2 MB	678,–
AS Soundsampler Maxi incl. Softw.	298,–
AS Soundsampler III incl. Software	588,–
Markendisk. 3.5" MF1DD 10St.	29,9
Markendisk. 3.5" MF2DD 10St.	34,5
Speichererweiterung auf 1 MB	a. A
ATARI-PC-Geh. m. Netzanschl.	149,–
Monitorumschalter o. Reset	79,–
P-Switch2 (2Drucker am ST)	188,–
P-Switch4 (b.4Drucker)	288,–
ST-Scanner incl. Software	229,–
ST-Scanner für STAD	199,–
G DATA Hardwareuhr o. löten	79,–
Druckerkabel 1,50 m	29,–
Floppykabel m. 2 Fl.-Stekcer	29,9
Drucker NEC P2200, dt.	998,–
Farbbandcassette P2200	17,5
Handy Scanner III m. Softw.	828,–
ST-Computer	auf Anfrag

NEU!: Public-Domain-Service

Rufen Sie uns an oder schreiben Sie uns:

Jürgen Vieth
Biesenstraße 75
4010 Hilden
Telefon 0 21 03 / 4 20 22
Katalog kostenlos

T.i.M Software

Time is Money

Aus unserer „Time is Money" Serie, die bequemste und preisgünstigste Art, Zahlungsvordrucke zu bearbeiten.
Neun verschiedene Formulare, Adressenverwaltung, Offene Posten, Schnittstelle zu TiM 1.1. Ein unerläßliches Hilfsmittel, für den Laien und Experten.

Banktransfer
Eine Verwaltung für Zahlungsträger



Detaillierte Informationen bei Ihrem Fachhändler oder direkt bei uns.

Unverbindliche Preisempfehlung:
Banktransfer DM 298,–;
Handbuch DM 30,–*;
Demo 10,–;
* wird bei Direktkauf angerechnet.

Händlerverzeichnis siehe Seite 68

C.A.$.H. GmbH
Robert-Bosch-Str. 20 a, 8900 Augsburg
Tel. 08 21 / 70 38 56

SIGNUM! ZWEI	355.-
OMIKRON.BASIC	150.-
STAD	149.-
TIMEWORKS	345.-
CALAMUS	819.-
PROTEXT ST 1.2	129.-
1st PROPORTIONAL	83.-
STEVE V.2.05	319.-
ASTERIX (m/f)	55.-
BAD CAT	49.-
BARDs TALE 1	73.-
DIZZY WIZZARD	53.-
FLIGHTSIMULATOR II (m/f)	119.-
JINXTER (m/f)	66.-
PSION CHESS (m/f)	63.-
STARTRECK	55.-
TERRORPODS	63.-

— Gratiskatalog anfordern —

SOFTWARE MICHAEL GRUBER
AM WEINGERT 27, D-8411 PETTENDORF
TEL.: 09409/2271 (24-h-Service)

ST-SPITZENSOFTWARE

ST-Firmenbuch: Doppelte Buchführung für jeden Selbständigen. **DM 139.-**

AS-Haushalt: Haushaltsbuchführung für alle Privathaushalte. **DM 99.-**

AS-Adress: Adressenverwaltung voll unter GEM, eigene Maskenerstellung. **DM 89.-**

AS-Diskop: Alle Funktionen des Desktop als ACC. Immer griffbereit. **DM 69.-**

Hardcopy+: Druckutility für alle 24-Nadeldrucker. Viele Optionen. **DM 49.-**

AS-Index: Erstellen von Indexverzeichnissen für 1st Word Plus Dateien. **DM 35.-**

ST-Printerfont: Druckerzeichensatzeditor für alle 24-Nadeldrucker. **DM 79.-**

Datenbankanwendungen: 8 Anwendungen für Adimens ST als fertige Definitionen. Lager, Adressen, Auftrag usw. **DM 89.-**

Gratisinfo 2/88 anfordern!!!

AS-Datentechnik Andreas Spitzbarth
Mainzer Str. 69 6096 Raunheim
☎ 06142 / 22677 o. 45879

SCANNER

für Atari ST an den Druckern: NEC P6
STAR NL 10, EPSON RX 80, FX 80, FX 85

Scannen Sie verwacklungsfrei durch den festen Sitz des Scankopfes. Für die genannten Drucker ist die Befestigung optimal realisiert. Die Leistungsmerkmale des Scanners:

- Anschluß über einen stabilen Modemstecker mit Gehäuse an der RS 232 Schnittstelle. *Kein* Öffnen des Rechners und *keine* Lötarbeiten erforderlich. Der empfindlichere Modulport bleibt frei. Uhrmodule, Epromkarten o.ä. müssen also nicht ständig aus- und eingesteckt werden.
- Die Software ermöglicht durch **bidirektionales** (!) Scannen eine Halbierung der Scanzeit (bei den Epson Druckern).
- Die Scanroutinen sind in Assemblercode geschrieben und garantieren ein Höchstmaß an Präzision.
- Justierung des Scankontrastes während des Scannens.
- Komfortable Einstellung von Scanparametern.
- Inverses Scannen und Zoomen ist möglich. Im Zoommodus wird die hohe Auflösung des Sensors ausgenutzt. Der NEC P6 z.B. kann bis zum Faktor 5 maßstabsgetreu zoomen.
- Grafikformate: Screen/Doodle- und Degasformat. Somit können die gescannten Grafiken problemlos mit den meisten s/w Zeichenprogrammen weiterverarbeitet werden. Auch die handelsüblichen Textprogramme lesen diese Formate.

SCANNER (fertig aufgebaut und getestet) mit Diskette, incl. ausführlicher deutscher Anleitung DM 298,- per NN zzgl. DM 9,- Versandkosten.

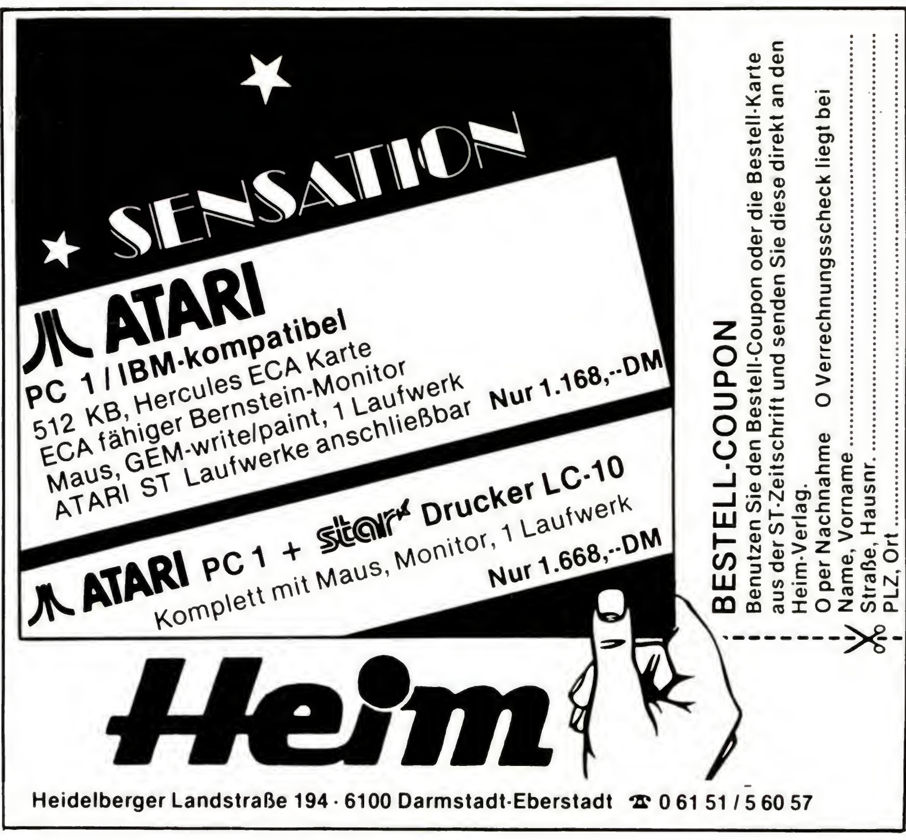
Dipl.-Ing. Gerhard Porada, Dürrlewangstr. 27
7000 Stuttgart 80, ☎ 0711 / 74 47 75.

SCANNER Hawk CP14 ST 1729.-
SIGNUM II DM 335.-
CALAMUS Desktop-P. DM 879.-
PD-Soft Atari-ST Hefte 1- 3.95.-
OCR — Scannersoftw. 898.-
CompTec L1 mit NEC 1037a 283.-
CL2 durchgeschleifter Port 315.-
GDATA Uhr 75.-Alles v. GDATA
CompTec Festplatte 20MB 998.-
Vortex HDplus 30/30MB 1339.-
Megamax MODULA II DM 329.-
STEINBERG Twenty four III 428.-
Steinberg Syntworks MT32 198.-
Alle Steinberg-Produkte erhältl
NEC P2200 dtsch. 12Mon.G. 958.-
NEC Multisync GS 3Aufl. 545.-
GRATIS - LISTE anfordern !!!!
CompTec GbR, Schulstraße 15
3584 Zwesten, Tel.: 05626/1374

Professionelle Finanzbuchhaltungsprogramme

Mini-Lern-Fibu 1.0	89,– DM
ST-Fibu 1,5	298,– DM
Mandantenfähig	398,– DM
ST-Fibu 2.0	498,– DM
Mandantenfähig	649,– DM

GEORG STARCK
Herzbergstr. 8 · D-6369 Niederdorfelden
Tel. 06101-3007 Hotline bis 22.00 Uhr




Heidelberger Landstraße 194 · 6100 Darmstadt-Eberstadt ☎ 0 61 51 / 5 60 57

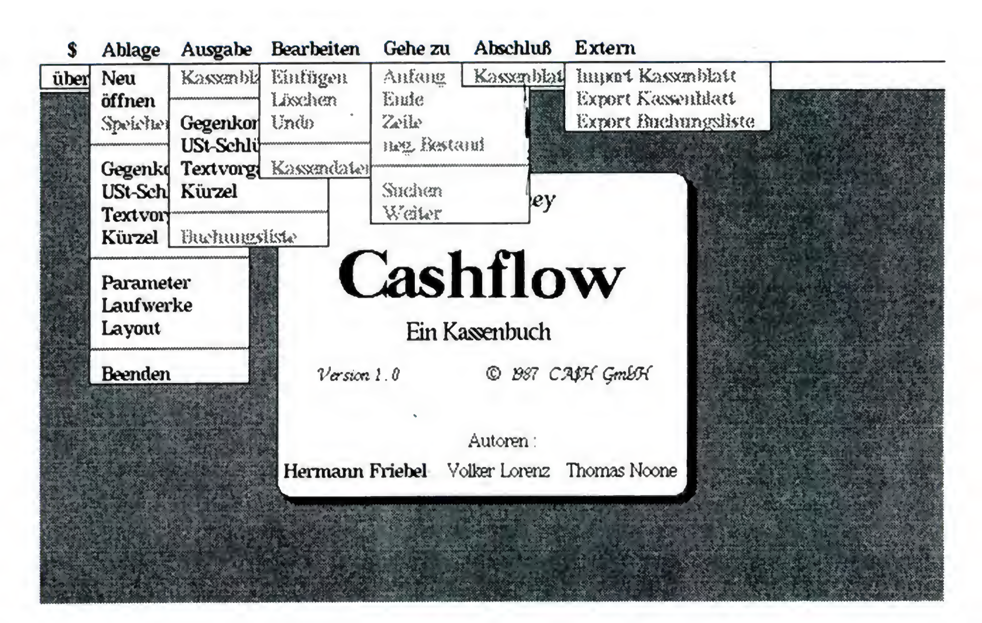
BARGELDFLUß

Seit kurzem ist von der Firma C.A.$.H. aus der Reihe "Time is Money" ein weiteres Produkt lieferbar. Wer hinter dem Titel "Cashflow" jetzt allerdings die Berechnung für den Mittelzufluß eines Unternehmens vermutet, irrt. Für dieses Programm ist der Titel eher wörtlich zu übersetzen, handelt es sich dabei doch um den Ersatz für ein herkömmliches Kassenbuch.

In der Reihe TiM sind derzeit vier Programme erschienen bzw. angekündigt. Das erste - T.i.M. Eine Buchführung - haben wir bereits im letzten Jahr in Heft 7/8 vorgestellt und getestet. Weitere Produkte sind noch das Kassenbuch, eine Verwaltung für Zahlungsvordrucke und eine Auftragsverwaltung. Das Kassenbuch ist seit einiger Zeit lieferbar und soll in diesem Artikel kurz beschrieben und getestet werden.

Bestseller

Jeder Selbständige, sei es ein Freiberufler, Handwerker, Händler oder Unternehmer, ist zu einer ordentlichen, d.h. von außen ohne Schwierigkeiten nachvollziehbaren Buchführung verpflichtet. Allerdings unterscheidet sich die Art und der Umfang der Buchführung je nach Größe des Umsatzes und der Unternehmensart. Allen gemeinsam dürfte aber ein bestimmter Anteil an Bargeldgeschäften sein, z.B. wird selbst ein Arzt seine Briefmarken noch in bar bezahlen (in welchem Postamt werden die Marken schon auf Rechnung, also unbar verkauft?). In jedem Fall wird also eine Form von Kasse geführt, sei es das Portomonaie (für eine ordentliche Buchführung die ungünstigste Art von Kasse...) oder die Hauptkasse.



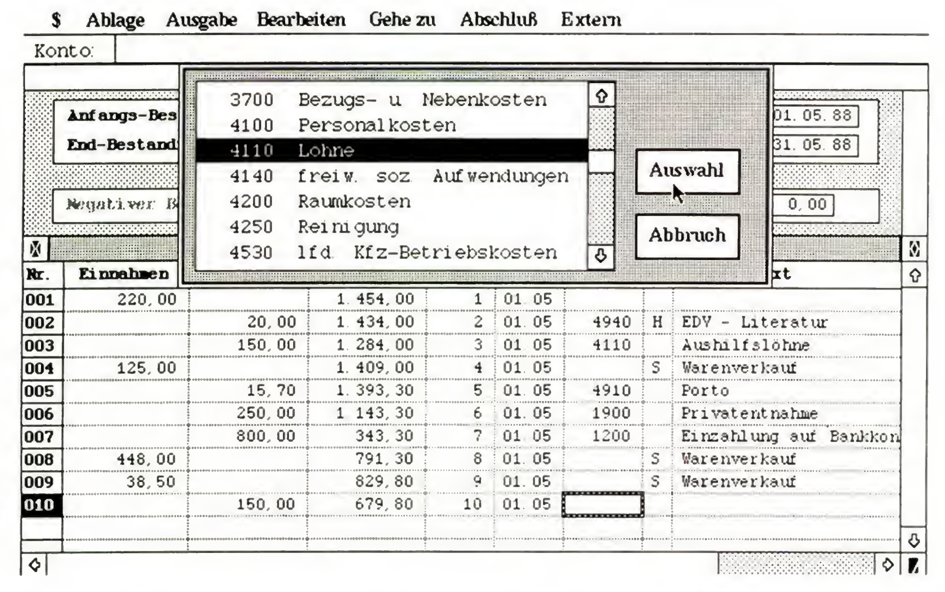
In das Kassenbuch werden mindestens einmal am Tag die Einnahmen oder Ausgaben mit den entsprechenden Belegen eingetragen, zusammen mit dem Übertrag vom Vortag muß die Differenz zwischen beiden Beträgen dann auch dem tatsächlichen Kassenbestand entsprechen. Diese Methode hat aber gewisse Nachteile. Abgesehen von einer wünschenswerten, getrennten Erfassung der verschiedenen Kosten und Erträge, entfällt die Möglichkeit der Bestandskontrolle bei Kassendifferenzen. Ein ausführliches Kassenbuch, in dem jede Bewegung von Hand eingetragen und berechnet wird, ist aber sehr aufwendig und macht zur besseren Unternehmensdurchsicht am Monatsende außerdem noch die Übertragung der Kosten und Ertragsdaten in die eigentliche Buchhaltung notwendig. Hier bietet sich ein computerunterstütztes Kassenbuch geradezu an.

Inhalt

Cashflow wird in der üblichen Plastiksoftbox geliefert. Man bekommt für DM 298,- die Programmdiskette und die Bedienungsanleitung - ein 82-seitiges DIN-A5-Heft. Die Anleitung ist in fünf Kapitel gegliedert, unter anderem gibt es eine Einführung in die Anwendung, die dem Cashflow-Laien gute Dienste leistet. Der spätere Nur-Benutzer wird durch die Kapitel "Beschreibung der Menüpunkte" und "Falls es Schwierigkeiten gibt" ausreichend und schnell unterstützt. Durch die gelungene Art der Seiteneinteilung - am linken Rand die Frage, wie sie im Inhaltsverzeichnis steht und rechts im Absatz die ausführliche Antwort - ist ein schneller Zugriff auf die Information auch innerhalb einer Seite gewährleistet. Der Text ist durch Schriftattribute deutlich in Information und Aktion getrennt; Bilder und Tabellen (die einen eigenen Index haben) leisten ihr übriges. Leider läßt der didaktische Aufbau einige Passagen - zumal für den Erstanwender - sehr mißverständlich werden; weiterhin vermindert ein schlechter, sinnverfälschender Seitenumbruch den sonst positiven Gesamteindruck.

Daten

Cashflow wird auf einer einseitigen Diskette geliefert und benötigt für den Betrieb den Monochrome-Monitor sowie einen Rechner mit derzeit mindestens 1 MByte-RAM. Das Pro-



gramm wird aber dahingehend überarbeitet, es wird in Kürze auch auf Rechnern mit kleinerem Speicher laufen. Cashflow ist mandantenfähig, pro Mandant sind bis zu 999 Kassenblätter und pro Kassenblatt bis zu 500 Einträge erlaubt. Für die spätere Übernahme der Kassendaten in die eigentliche Buchhaltung kann jetzt schon ein Kontenrahmen mit bis zu 2000 Konten definiert werden. Für ein bequemes Arbeiten sind Hilfsdateien für Texte, Kürzel und Umsatzsteuerschlüssel einrichtbar; sie erlauben jeweils bis zu 100 Einträge. Als Besonderheit bei Cashflow sei hier noch der Import bzw. Export von Dateien genannt, die das Programm an ein beliebiges Buchhaltungsprogramm anklinken lassen. Ideal ist hier natürlich die Kombination mit dem Programm aus der Reihe TiM.

Das Programm benutzt eine eigene Schriftform, so daß das Erscheinungsbild am Anfang eher an einen MacIntosh als an einen ATARI erinnert. Abgesehen von ästhetischen Gesichtspunkten hat dieser neue Font (er wird dann auch von den Accessories benutzt) einen ganz praktischen Hintergrund: auf dem Bildschirm lassen sich jetzt 96 Zeichen pro Zeile darstellen. Nach dem Start müssen zuerst die Vorgabedateien und der Kontenrahmen erstellt werden, auf welche das Programm zugreift. Dabei kann bereits der Kontenrahmen von TiM verwendet werden. Pro Mandant werden über ein Arbeitsblatt die Kassendaten eingegeben. Hier wird u.a. der Istbestand und neben der Adresse auch die gewünschte Auswahl an Vorgabedateien eingetragen. Danach steht das Kassenblatt zur Bearbeitung zur Verfügung. In jeder Zeile kann nach der vorgegebenen Zeilennummer die Einnahme oder Ausgabe (wird anschließend sofort berechnet und in das Feld "Bestand" eingetragen), die Belegnummer (auf Wunsch automatisch), das Datum (ebenfalls automatisch möglich), das Konto, der Steuerschlüssel und der Buchungstext eingetragen werden. Zusätzlich können für die Felder Konto, Steuerschlüssel und Buchungstext die Hilfsdateien eingeblendet werden. In diesen Dateien kann über einen Rollbalken beliebig geblättert und die Einträge über einen Doppelklick der Maustaste übernommen werden. Alle eigenen Eingaben erfolgen vorab in einer Editzeile am oberen Bildschirmrand; erst nach Drücken der Entertaste wird der Inhalt dieser Zeile in das eigentliche Feld übertragen. Der Abschluß eines Eintrages erfolgt endgültig über >Return<. Solange sich ein Kassenblatt in Bearbeitung befindet, kann jederzeit auf die Einträge zugegriffen werden, Korrekturen sind ebenfalls möglich. Erst nach dem Kassenblattabschluß kann dieses nur noch ausgegeben werden.

Eine der bemerkenswerten Eigenschaften dieses Programms sind die Ausgabemöglichkeiten. Dabei wird nicht einfach stur die Liste präsentiert, sondern über eine wirklich sehr umfangreiche Vorgabe- und Abgrenzungsmöglichkeit eine gezielte Suche oder sortierte Ausgabe erreicht.

Start

Cashflow ist nicht kopiergeschützt und kann somit ohne Probleme auch mit der Festplatte benutzt werden. Lediglich am Anfang muß einmal das Laufwerk oder die Partition für die Arbeits- und die Archivdateien eingestellt werden. Als Drucker sollte ein Exemplar mit 136 Zeichen zur Verfügung stehen, um die Listen übersichtlich drucken zu können. Allerdings macht auch ein normaler Drucker mit 80 Zeichen keine Probleme, zumal über einen Initialisierungsstring im Menüpunkt >Layout< der Drucker eingestellt werden kann.

$ Ablage Ausgabe Bearbeiten Gehe zu Abschluß Extern

Text: Aushilfe Hr. Jemand (Inventur 30. 4.)

KASSENDATEN

Anfangs-Bestand: 1 234, 00 — Kassen-Konto: 1000 — Datum von: 01. 05. 88
End-Bestand: 679, 80 — Einträge: 10 — bis: 31 05. 88

Negativer Bestand ab Eintrag-Nr.: — Mindestausgleich: 0, 00

Kassenblatt: TEST. 001

Nr.	Einnahmen	Ausgaben	Bestand	BelNr	Datum	Konto	S	Text
001	220, 00		1 454, 00	1	01. 05			
002		20, 00	1 434, 00	2	01. 05	4940	H	EDV - Literatur
003		150, 00	1 284, 00	3	01. 05	4110		Aushilfslohne
004	125, 00		1. 409, 00	4	01. 05		S	Warenverkauf
005		15, 70	1 393, 30	5	01. 05	4910		Porto
006		250, 00	1. 143, 30	6	01. 05	1900		Privatentnahme
007		800, 00	343, 30	7	01. 05	1200		Einzahlung auf Bankkon
008	448, 00		791, 30	8	01. 05		S	Warenverkauf
009	38, 50		829, 80	9	01. 05		S	Warenverkauf
010		150, 00	679, 80	10	01. 05	4110		

$ Ablage Ausgabe Bearbeiten Gehe zu Abschluß Extern

Suchen

Nr. Einnahmen Ausgaben Bestand BelNr Datum

100,00 50,00

Konto USt Text Kürzel

< kleiner = gleich > größer Suchrichtung vorwärts rückwärts

Suchen Beenden Abbruch

Ein Beispiel dieser Wahlmöglichkeiten sehen Sie im nebenstehenden Bild. Die Suche nach Einträgen oder Differenzen wird dadurch sehr kurzweilig. Negative Bestände der Kasse, die es ja eigentlich nicht geben kann und nur aufgrund von falschen bzw. fehlenden Buchungen zustande kommen, werden im normalen Eingabemodus gesondert angezeigt und können in der Auswahl direkt aufgerufen werden.

Meinung

Die vielen Feinheiten, die während der Arbeit mit diesem Programm gefunden wurden und hier gar nicht alle berücksichtigt werden können, zeugen von einem durchdachten Konzept, welches hinter diesem Programm zu steht. Als Beispiele hierfür sind der Auf-/Abwärtszähler für das Datum über die linke und rechte Maustaste oder die Reaktivierung von bis zu 50 versehentlich gelöschten Buchungen über >Undo< zu nennen. Das Programm ist schnell und anwenderfreundlich. Durch die konsequente Betreuung seitens der Firma C.A.$.H., welche eine kostenlose Hotline für die Benutzer anbietet und deren Benutzung auch wünscht, wird eine ständige Verbesserung und Weiterentwicklung ermöglicht. Auch sind in Cashflow bereits die Erfahrungen und Kundenwünsche aus der Entwicklung der TiM-Buchhaltung (inzwischen ist z.B. direktes Anspringen der Felder über den Mauszeiger in beiden Programmen möglich) eingeflossen.

Für mich persönlich bleiben nur ein paar kleine Kritikpunkte. So wäre es wünschenswert, wenn die Hilfsdateien alphanumerisch sortiert werden könnten; gerade der Erstbenutzer kann am Anfang nicht alle Vorgaben kennen - dadurch sehen diese Dateien zum Schluß aus wie Kraut und Rüben (zumindest während meiner Testphase). Auch sollte die Auswahlzeile nicht immer wieder an den Anfang der Hilfsdatei zurückspringen; gerade in einer Kassendatei beschränken sich die Buchungen in der Hauptsache auf ein paar wenige Vorgaben (frei nach Murphy liegen die dann bestimmt am Ende). Und da man auch in der Regel erst bei der Bearbeitung das Fehlen eines Vorgabetextes bemerkt, wäre eine direkte Übernahmemöglichkeit aus der Editzeile in die Vorgabedatei sehr wünschenswert.

Alles in allem kann ich Cashflow dem Anwender, der sowieso ein Kassenbuch führen muß und eine Übernahme der Daten in seine Buchhaltung wünscht, nur empfehlen - zumal das Programm mit DM 298,- seinen Preis wert ist.

hp

Cashflow - Ein Kassenbuch
Preis: DM 298,00

C.A.$.H. GmbH
Schillerstraße 64
8900 Augsburg
Tel. 08237/1020

$ Ablage Ausgabe Bearbeiten Gehe zu Abschluß Extern

Ausgabe Kassenblatt — Selektion ein — Sortieren ein

Nr. Einnahmen Ausgaben Bestand BelNr Datum

untere Grenze
obere Grenze
Sortier-Rang
auf-/absteigend

Konto USt Text Kürzel

untere Grenze
obere Grenze
Sortier-Rang
auf-/absteigend

Bildschirm Drucker Datei Abbruch

ENDE

MUSIKMESSE FRANKFURT

Wie jedes Jahr machte sich auch dieses Jahr das erwartungsvolle Messekribbeln breit, das im Innern des wahren Musikers jedes Jahr die Musikmesse Frankfurt ankündigt. Musikmarkt live. In Superlative. Jedes Jahr größer und vor allem lauter, was auch die mit Phonmeßgeräten ausgestatten Streifen nicht verhindern können. Das Getöse ist aber auch wirklich beeindruckend - weite Hallenflure gefüllt mit Ständen, deren jeder den anderen mit seinem akustischen und optischen Erscheinungsbild zu übertreffen sucht, kurz, die totale Reizüberflutung. Aber schön ist's doch trotzdem immer wieder.



Wie sich das gehört, war auch die 9. Musikmesse in Frankfurt wieder größer als alle bisherigen. Insgesamt 38 Länder und 890 Aussteller führten ihre Produkte vor, vom Notenpapier zu Kontrabaßsaiten, vom Konzertflügel zur Blockflöte, vom Synthesizer zum Studiomischpult - und natürlich jede Menge Musiksoftware.

Mit großer Befriedigung darf der Atari-Fan zur Kenntnis nehmen, das kein Rechner so weite Verbreitung auf dem Musikmarkt gefunden hat, wie der Atari ST. Vom Sequencer zur Mischpultautomation, Lichtprogrammen und was man sonst noch so mit einem Computer machen kann, kaum ein Stand auf der Musikmesse, der etwas anderes als rein akustische Instrumente vorzuführen hatte, ließ nicht an irgendeiner Ecke seines Standes einen Atari sehen. Für den europäischen Markt gilt also immer noch und immer deutlicher: Der ST ist der Musikrechner Nummer 1, trotz Preissenkungen bei Apple und der Flut billiger IBM-kompatibler sind die entsprechenden Maschinen keine Konkurrenz für den ST.

Inzwischen ist es wirklich so weit, daß man ein Studio mit einer ganzen Atari-Sammlung zustellen könnte, jeder mit einer anderen Aufgabe. Dem wahren Atari-Fan sollte das zu tiefer Befriedigung gereichen, allein schon deshalb, weil Hauptkonkurrent Amiga immer noch so gut wie nichts von sich sehen ließ.

Trotz extremer Ausstellerzahl: Große Sensationen gab es nicht auf der Musikmesse '88. Fast alle ausgestellten Geräte waren in der Fachpresse bereits seit einiger Zeit angekündigt. Der größte Knüller aus der Sicht der Computer-unterstützten Musiziererei ist vielleicht der große Bösendörfer-Flügel, dessen Mechanik mit Lichtschranken abgetastet wird und dessen Tasten sich auch mit einer komplizierten Magnet-Mechanik per Computer steuern lassen. Kurz, dieser Flügel ist die totale Steigerung der alten mechanischen Klaviere, die über Papierrollen gesteuert wurden. Ein an das Instrument angeschlossener Computer kann alle auch sehr feinen Nuancen des Tastenanschlags aufzeichnen und wiedergeben. Gera-

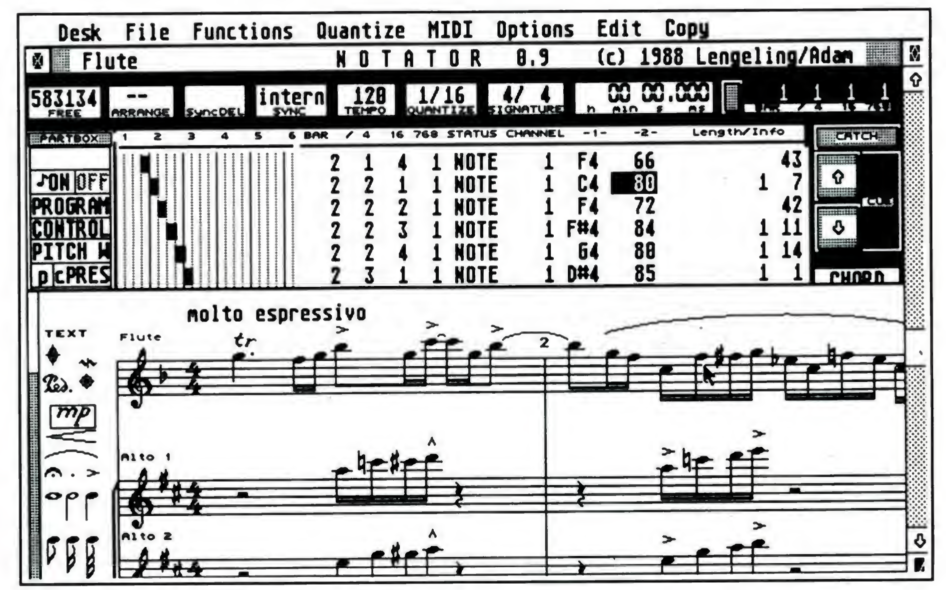
de auch als Produktionswerkzeug für Klavieraufnahmen ist dieser Flügel interessant, einziger Nachteil: Der Preis ist wahnsinnig hoch, da das System in einen großen Konzertflügel eingebaut wird und auch selbst wohl nicht gerade billig ist. Um die 200.000,- DM muß man für ein solches Instrument bezahlen.
Also machen wir uns an die Arbeit:

Was gibt's alles neu?

Die Hamburger Firma **C-Lab** hat auf der Musikmesse ihren lange erwarteten 'Notator', eine Kombination aus dem seit langem bekannten, sehr leistungsfähigen Sequencer 'Creator' und einem Notendruckprogramm, das besonders durch seine Geschwindigkeit beeindruckt, vorgestellt. Die maximal 64 Aufnahmespuren des Creators können mit diesem Programm in Notenschrift umgesetzt, bearbeitet und auch gedruckt werden. Die Geschwindigkeit ist wirklich verblüffend: Man kann während einer Aufnahme die Noten sofort auf dem Bildschirm erscheinen sehen. Natürlich können auch Sonderzeichen und Text in das Notenbild eingefügt werden. Mit der Maus können direkt Noten in das Notenbild eingefügt oder daraus gelöscht werden, wobei alle Änderungen natürlich auch hörbar sind. Man kann den Notator also auch einfach wie einen Creator mit zusätzlichem Noteneditor benutzen. Dabei ist besonders angenehm, daß man mehrere Spuren gleichzeitig im Notenbild sehen und edieren kann (in verschiedenen Systemen). Die Aufteilung des Bildschirms zwischen Grid-Editor und Noten ist dabei frei wählbar. Raffiniert.
Soweit man das nach dem kurzen Messe-Blick sagen kann, ist die automatische Umsetzung der Musik in ein Notenbild, wie sie der Notator durchführt, sehr gelungen, was man automatisch tun kann, scheint das Programm auch richtig zu machen, leider sind jedoch noch keine Funktionen vorhanden, die die Bearbeitung von automatisch nicht lösbaren Problemen (wie z.B. komplexe Polyphonie oder Fähnchen/Balkensetzung für Vokalnoten oder in komplexen Taktarten) erlaubten. Aber was nicht ist, wird sicherlich kommen, wofür gibt es Updates. Die Druckqualität ist jedenfalls gut. Leider ist der Notator nicht ganz billig: Fast 1000,- DM muß man für das Programm bezahlen, für Creator-Besitzer gibt es einen Update-Preis.
Für die C-Lab-Sequencer gibt es auch eine Midi-Output-Erweiterung namens 'Export', die dem Atari für verhältnismäßig wenig Geld zu drei zusätzlichen Midi-Outputs verhilft. Damit dürften Timing-Schwankungen wegen Überlastung des Midi-Ausgangs der Vergangenheit angehören.

C-Lab-Software,
Postfach 700303
2 Hamburg 70

Bei der Hamburger Konkurenz, bei **Steinberg**, gab es vor allem eine neue Version des TwentyFour Sequencers



Screenshot des C-LAB-Notators

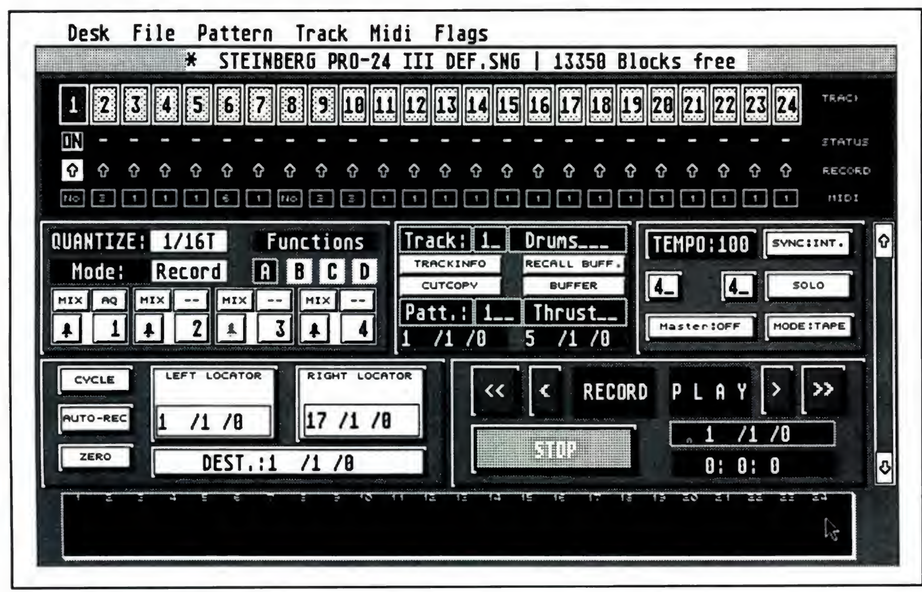
REPRODUKTION EINES ORIGINAL AUSDRUCKS VOM C-LAB NOTATOR (NEC P6)

molto espressivo

text styles! small underlined italic Mixed

Originalausdruck des C-LAB-Notators

zu sehen, die den Namen TwentyFour III trägt. Mit dieser Namenserweiterung wollte man wohl den großen Schritt dokumentieren, den der TwentyFour mit der neuen Version macht. Das Programm enthält jetzt eine völlig neue Aufnahme-Methode, die es möglich macht, auf bis zu vier Spuren gleichzeitig aufzunehmen. Außerdem kann die Aufnahme bereits mit vielseitigen Werkzeugen bearbeitet werden, ohne daß man unterbrechen muß. Quantisieren ist genauso möglich wie das Löschen aller oder auch nur ganz bestimmter Töne der Aufnahme, natürlich einzeln für jede der vier simultan aufnehmbaren Spuren. Überhaupt wurde



Die neue Version 3.0 des Twenty Four

Der Drum Editor des Twenty Four 3.0

die Bedienung des Sequencers stark vereinfacht und übersichtlicher, zum Beispiel sind jetzt die Midi-Kanaleinstellungen jeder Spur ständig sichtbar. Das Programm kann jetzt auch fast vollständig von einem Midi-Keyboard ferngesteuert werden, so daß man die Hände bei einer Produktion praktisch nie mehr vom Keyboard wegnehmen muß. Aufnehmen, bis das Stück fertig ist, schnell und ohne dauernde Unterbrechungen. Der Noten- und Grid-Editor haben Gesellschaft bekommen, der Drum-Editor funktioniert wie ein extrem komfortabler Drumcomputer. Sie können Noten mit der Maus setzen und Löschen oder auch einspielen, ganz wie im 'richtigen Leben'. Dazu können Sie bis zu drei komplette Drumkits mit insgesamt 96 Instrumenten definieren.

Der TwentyFour III kann Stücke im Standard-Midi-File-Format lesen und schreiben. Dieses Format, das sich immer mehr durchsetzt, soll dem Austausch von Stücken, die mit verschiedenen Sequencern gemacht wurden, dienen. Neben vielen weiteren verbesserten Details gibt es auch eine neue Anleitung.

Eine sehr gute Idee ist das 'Switcher'-Programm, das es erlaubt, bis zu zehn Programme gleichzeitig im Speicher zu halten. Dabei kann jedem Programm eine bestimmte Speichergröße zugewiesen werden. In Zusammenarbeit mit dem Steinberg Key-Expander ist es auch möglich, zwischen mehreren Programmen, die durch ein Key-Modul kopiergeschützt sind, umzuschalten, wobei Steinberg-Programme sogar automatisch 'ihren' Key finden können. Ein Test dieses universellen Switcher-Programms, das nicht nur Steinberg Programme verwaltet, findet sich in einer der nächsten Ausgaben.

Auch das Notendruckprogramm Masterscore ist jetzt in einer neuen Version erhältlich. Die Bedienung wurde verbessert und ist jetzt komfortabler. Leider ist die Behandlung von Polyphonie immer noch fehlerhaft. Sehr groß ist der Sprung zur neuen Version daher nicht.

Einige neue Editoren wurden ebenfalls vorgestellt: Ein Editor aus der Synthworks-Reihe für den Roland MT32, sowie ein neuer, extrem leistungsfähiger Editor für den Emax, ein Samplingkeyboard der Firma E-Mu.

Schließlich gibt es die Mischpult-Automation für das Yamaha DMP-7 Pult jetzt auch für den Mac II, sowie eine neue Atari-Automation für TAC-Pulte. Der Steinberg Tape-Controller erlaubt es, Tonbandmaschinen aus dem Sequencer heraus fernzusteuern. Im Augenblick funktioniert das nur mit Fostex-Geräten, aber das soll sich in Zukunft ändern.

Das integrierte Editorsystem X-Syn, das früher von Beam Team vertrieben wurde, ist von Steinberg übernommen worden.

TSI,
Neustr. 12
5481 Waldorf

Nur wenige Neuigkeiten gab es bei **Hybrid Arts**. Das ADAP-Sound Sampling-System scheint jetzt endlich so weit gediehen zu sein, daß man mit der Software arbeiten kann. Inzwischen sind auch alle wesentlichen wichtigen Funktionen integriert, so vor allem eine sehr gut gelungene Auto-Loop-Funktion.

Eine eigene Harddisk-Serie wird es von Hybrid Arts geben, die große Kapazitäten zu extrem günstigen Preisen bietet. Wer eine Harddisk sucht, sollte sich um ein Angebot bemühen.

Entwickelt wurde die Harddisk-Serie vor allem für das ADAP II Soundrack, das in der Lage sein soll, direkt auf eine Festplatte zu sampeln. Damit erhält man, entsprechende Harddisk-

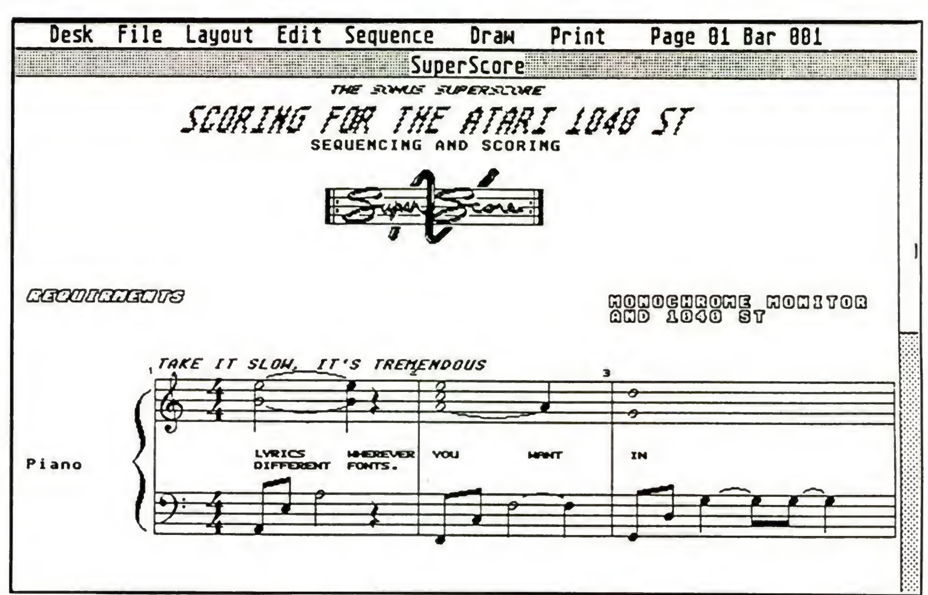
Größen vorausgesetzt, ein digitales Mastering-System, mit dem auch digitales Schneiden zu günstigen Preisen möglich sein wird. Auf der Messe gab es einen Prototyp zu sehen, der bereits Aufnehmen und Wiedergeben konnte.

Nicht völlig neu, aber dafür weiter verbessert präsentierte sich das 'EZ-Score Plus' Notendruckprogramm. Die Ausdruckqualität wurde im Vergleich zur ersten Version stark verbessert. Einen Test finden Sie im nächsten Heft.

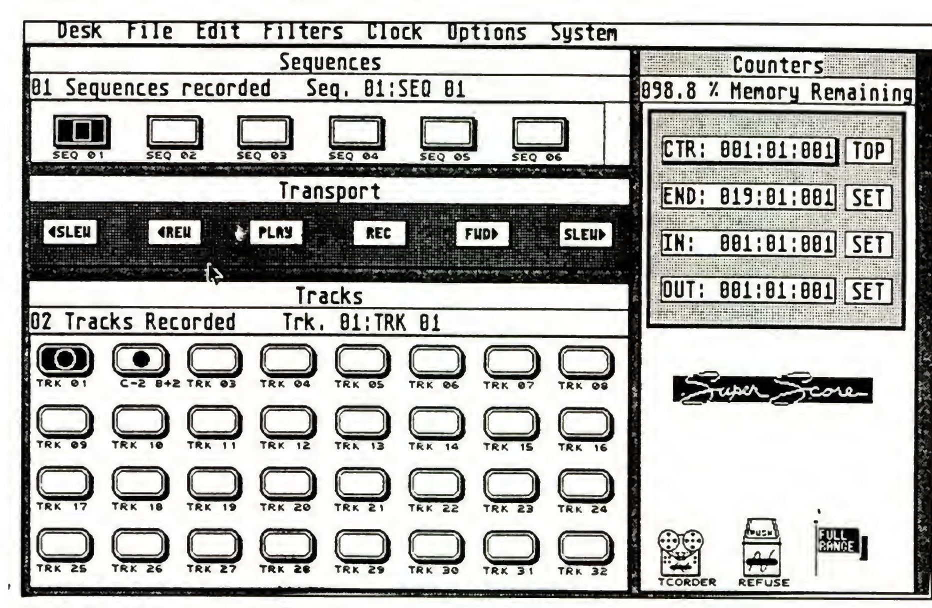
Für August wurde eine voll professionelle Version des EZ-Score angekündigt, auf die man gespannt sein darf. Bis zu 60 Systeme werden auch große Partituren erlauben.

Ein Switcher-Programm mit dem Namen Hybri-Switch dient dazu, mehrere Programme im Speicher zu halten und schnell zwischen ihnen umzuschalten. Maximal 16 Programme kann das Programm verwalten, leider arbeitet es nur mit Hybrid Arts Produkten zusammen.

Hybrid Arts Deutschland,
Lindenscheidstraße 1
6230 Frankfurt/Main 80



Umfangreiche Notendruck-Möglichkeit bietet der SuperScore von Sonus



Der 32-Track-Sequencer des SuperScore

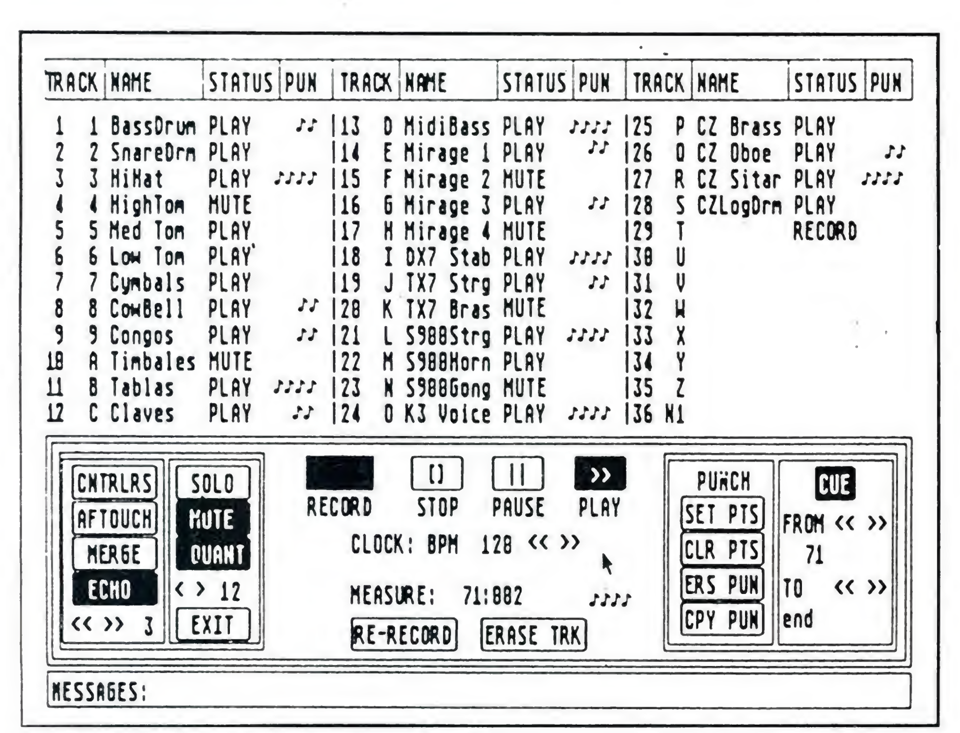
Das erste Mal auf einer Musikmesse war die amerikanische Firma **SONUS** vertreten. Unter ihren Produkten, die Sequencer, Sound-Editoren sowie diverse Midi-Hardware-Erweiterungen für den MacIntosh umfassen, fiel besonders das Notendruckprogramm 'SuperScore' auf.

Dieses Programm besteht aus einem 32-Spur Sequencer, es ermöglicht 24 verschiedene Sequenzen (mit jeweils 32 Spuren) aufzunehmen und zu arrangieren. Die üblichen Editiermöglichkeiten wie Quantisieren und Transponieren usw. sind selbstverständlich vorhanden. Hat man das Stück zu seiner Zufriedenheit eingespielt, begibt man sich auf die Score-Writer-Seite, wo die Aufnahme in Notenschrift transkribiert wird. Hier bestehen nun weitgehende Layout-Möglichkeiten, so daß auch polyphone oder Vokalmusik, mit der die automatische Transkribierung zwangsweise überfordert sein muß (schließlich besitzt der Computer ja kein musikalisches Verständnis) vernünftig notiert werden kann. Zusätzlich stehen jede Menge Sonderzeichen sowie ein komplettes objektorientiertes Zeichenprogramm zur Verfügung. Text kann sowohl frei als auch speziell für Liedtexte (das heißt automatisch unter Noten zentriert) positioniert werden. Noten können selbstverständlich auch per Maus eingesetzt werden, auch Perkussions- und Flageolettnotenköpfe sind vorhanden, kurz alles, was das Herz oder das Notenlayout begehrt. Bisher steht leider nur ein Druckertreiber für 9-Nadeldrucker zur Verfügung, in Kürze soll das Programm mit stark verbesserter Druckqualität auch 24-Nadel- und den Atari-Laserdrucker unterstützen. Auch PostScript-Ausgabe ist vorgesehen. Zum Erscheinen dieser Ausgabe soll es bereits eine erheblich komfortablerer Version des Programms, dessen Preis in Deutschland bei 645,- DM liegen wird, geben. Für den ST gibt es noch zwei Sequencer namens Masterpiece und SST, die ähnlich aufgebaut sind, wie der im SuperScore integrierte Sequencer, allerdings um zusätzliche Möglichkeiten erweitert wurden.

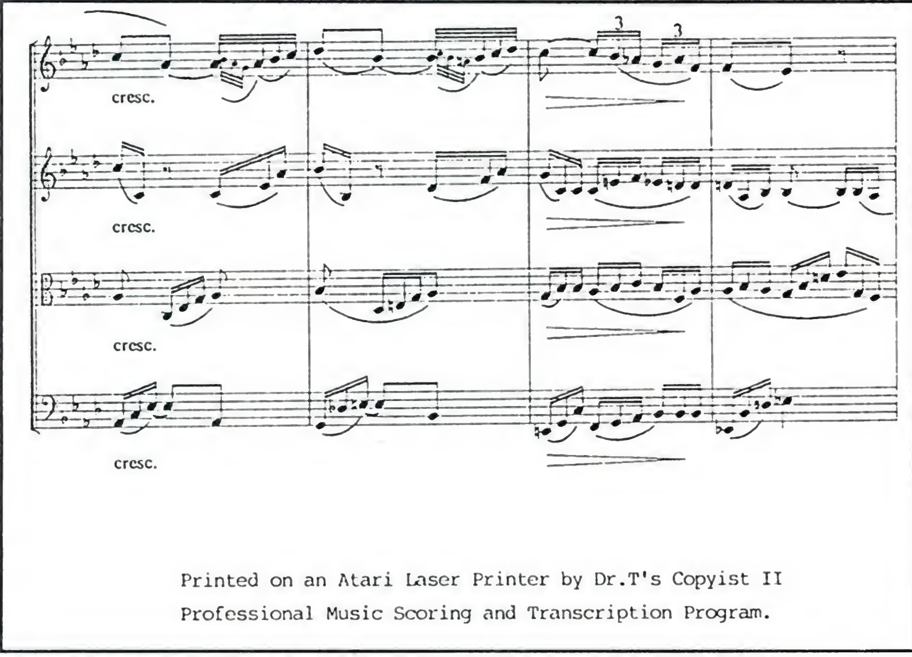
Interessant ist bei allen Sonus-Sequencern die Möglichkeit, mit einem preisgünstigen Hardware-Adapter (unter 150,- DM) einen zweiten parallelen Midi-Ausgang an den Atari anzuschließen.

Es existieren schließlich noch Editoren für den Ensoniq-Mirage Sampler und diverse Synthesizer (D-50, ESQ-1 und SQ-80).

Ein interessantes Hardware-Produkt ist der SMPTE/Midi-Timecode-



Der Keyboard Controlled Sequenzer von Dr. T's



Probeausdruck mit dem ATARI Laser

Umsetzer SMX 2000, der Timecode-Synchronisierung für weniger als 1000,- DM erlaubt.
Auch für IBM-Kompatible und den Apple-MacIntosh gibt es eine Reihe von Programmen, vom Sequencer über Editoren bis hin zur Mischpultautomation.
Nähere Informationen zu den SONUS-Produkten sind über die Redaktion der ST-Computer zu erfahren.
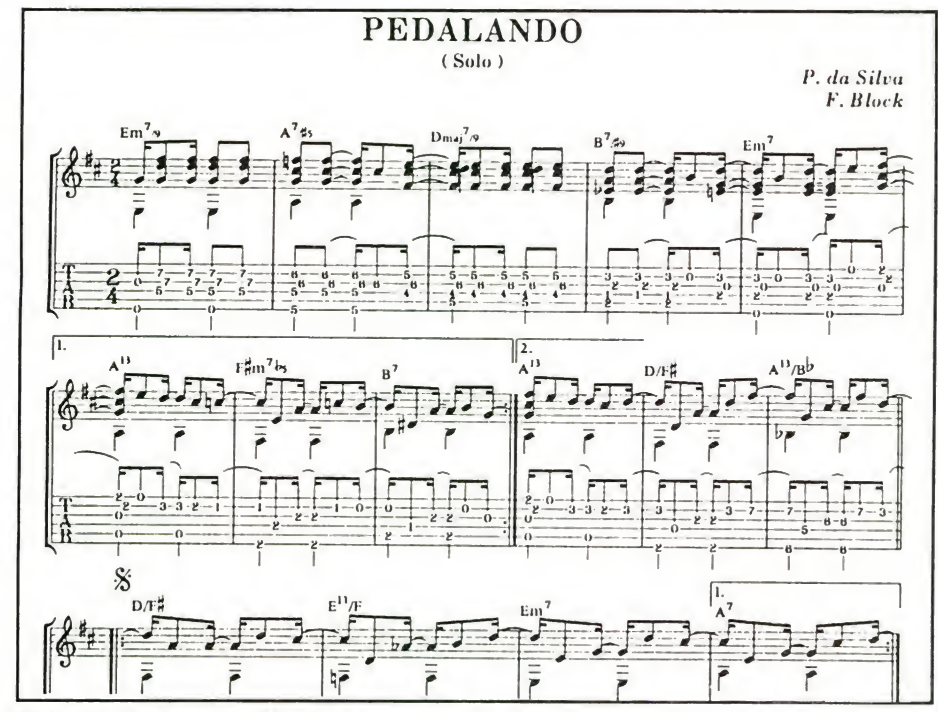
Dr. T's stellte eine neue Version des Keyboard Kontrolled Sequencers (kurz KCS) aus, nämlich Version 1.6 Level II. Diesen Sequencer gibt es jetzt auch in Versionen für den Apple MacIntosh und den Commodore Amiga. Auf dem ST ist das Programm jetzt auch kompatibel mit dem Dr.T's Multi Programming Environment, einer speziellen Software, die es erlaubt, mehrere Dr. T's Programme gemeinsam miteinander laufen zu lassen. Dabei können die Programme sogar miteinander kommunizieren. Zum Beispiel kann der Sequencer laufen, während man mit einem Editor Sounds an einem Synthesizer verändert.

Mitgeliefert wird auch ein Kompositionshilfe-Programm, das den Namen 'Programmable Varations Generator', kurz PVG, trägt. Dieses Programm ist in der Lage, aus vorhandenem melodischen oder harmonischen Material Variationen zu erzeugen. Man kann dabei alle Parameter der Veränderungen selbst bestimmen. Auch das Notendruckprogramm 'The Copyist' liegt in einer erweiterten Version vor, die jetzt auch Laserdrucker und PostScript unterstützt. Außerdem kann das Programm Dateien an die wichtigsten Desktop Publishing Programme übergeben, so an den PageMaker, den Ventura-Publisher und den neuen Publishing Partner Pro für den ST.
Mit den neuesten Erweiterungen enthält Dr. T's Programmserie wirklich alles, was man zum Musikmachen braucht. Mit dem MPE lassen sich darüber hinaus alle diese Programme zu einem großen Musiksystem zusammenfassen.

MEV,
Karl-Hromadik-Str. 3
8000 München 60

Apropos Notendruck: Das professionelle Notensatz-System Amadeus der Münchner Firma **Amadeus Music Software**, das bisher den Atari ST nur als intelligentes Terminal verwendet, wird in Kürze auch vollständig auf dem Atari lauffähig sein (Als Einplatz-Singel-User-System).
Anschließbar ist alles vom Matrixdrucker bis zur Fotosatzanlage. Von Preis und Leistung dürfte das Programm ausschließlich für professionelle Anwender in Frage kommen: Es kostet um die 70.000,- DM, kann aber absolut druckreife Vorlagen produzieren. Einen eigenen Fotosatz-Belichter hat die Firma ebenfalls im Programm. Dieser Belichter wird über ein Spezial-Interface an den Atari angeschlossen und soll auch für OEM's zu einem sehr günstigen Preis verfügbar sein.

Amadeus Musik Software Kurt Maas
Rohrauer Straße 50
8000 München 71



High price, high quality: Das Notensatzsystem von Amadeus

Bei **G.C. Geerdes** gibt es eine komplette Palette von Editoren für so ziemlich jeden halbwegs verbreiteten Synthesizer. Besonders interessant ist ein Sample-Editor, der sich 'Sample Construction Set' nennt. Zuerst einmal wird dieser Editor für den PSE-900 veröffentlicht, aber auch bald an andere Instrumente angepasst werden. Das Programm wird extrem komfortabel sein und enthält auch eine Möglichkeit zur harmonischen Analyse mit bis zu 64 Harmonischen. Dabei können auch Oberwellen verschiedener Samples ausgetauscht werden. Digitales Filtern usw. ist ebenso möglich.
Auch ein Midi-Tool, das als Accessory installiert wird, gibt es. Dieses Tool enthält einen Midi-Monitor, ein Masterkeyboard- und ein Dump-Utility.

G.C. Geerdes,
Guerickestr. 43
1000 Berlin 10

Die französische Firma **Digigram** bietet ein Orchestral Composer-Programm namens 'Big Band' an. Das Programm kann zu einer Melodie vom Rhythmus bis zur zweiten Stimme so ziemlich alles generieren, was man in einem Arrangement braucht, und das in verschiedenen Stilrichtungen. Die Vorführungen der Software waren recht interessant, wieviel diese Composer-Software wirklich kann, muß ein ausführlicher Test zeigen.
Auch ein einfacher 24-Spur Sequencer ist bei der Firma im Programm.

COMUS France
67 Boulevard J.B. Oudry
94035 Creteil Cedex France

Einen weiteren, sehr interessanten Sequencer gibt es von **Hitec**. Das 'Midas' getaufte Programm bietet 255 Spuren bei einer internen Auflösung von 1/384tel Note. Die Anzeige von Spureninhalt und Midi-Events ist sehr übersichtlich, auch die Ediermöglichkeiten sind sehr leistungsfähig.

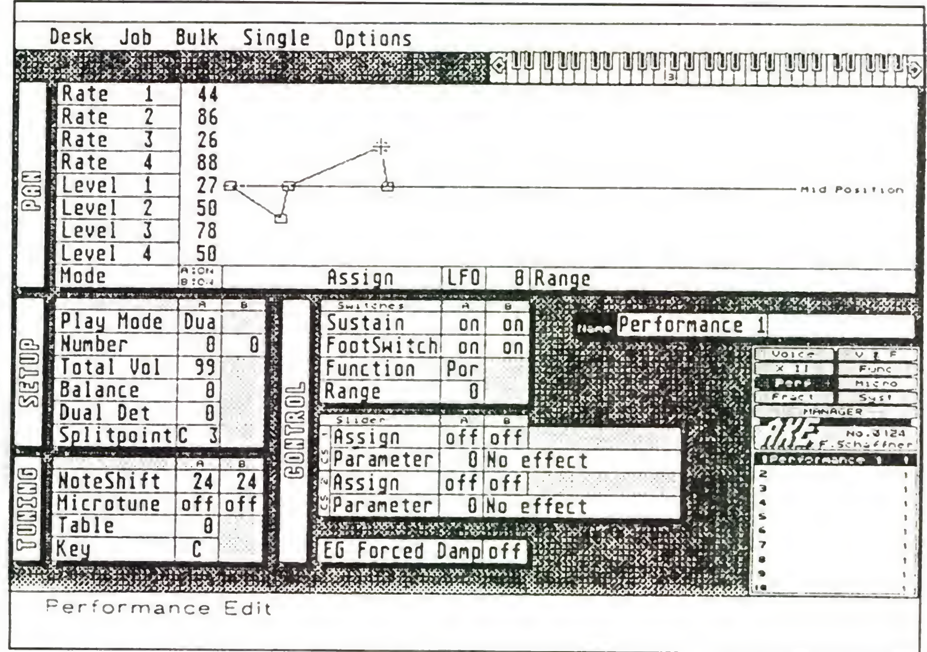
Hitec Musikelektronik,
Unterkessacher Str. 35
7109 Widdern

Passport hat eine einfache und billige Version des MasterTracks Pro-Sequencers auf den Markt gebracht, der MasterTracks Jr. (für Junior) heißt. Das Programm ist mit dem großen Bruder kompatibel.

MEV,
Karl-Hromadik-Str. 3
8000 München 60

Aus Darmstadt kommt ein neues DX-7 Editor-Programm. Es heißt 'AXE' und kommt von der Firma **Midi-Soft.** Die Darstellung der Parameter erfolgt grafisch und ist sehr übersichtlich. Das Programm kann auch Sequenzen des Steinberg TwentyFour Sequencers abspielen.

Verleihfix Midi-Soft
Wiener Str. 59
6100 Darmstadt



Der DX-7 Editor "AXE" von Midi-Soft

Soft Arts aus Berlin bietet Editoren für eine Reihe weit verbreiteter Synthesizer an, unter anderem ein komfortables Sample-Editor-Programm für den Korg DSM-1.

Soft Arts
Postfach 440202
1000 Berlin 44

Editoren für Roland, Korg und Ensoniq-Synthesizer, sowie einen Sample-Editor für Korg-Sampler gibt es bei PA-Decoder.

PA-Decoder,
Osterfeldstr. 1
4520 Melle 9

Unter dem Label 'LowLands' gibt es auch eine Editor-Serie aus den Niederlanden, die von der Firma **Mopro** vertrieben wird. Das interessanteste Programm aus dieser Serie scheint ein Midi-Interpreter zu sein, der besonders für Experimente geeignet sein dürfte. Ähnlich wie bei einem Basic-Interpreter, können Midi-Kommandos mit einigen Hilfsinformationen wie Tempo usw. programmiert werden.

Mopro,
P.O.Box 2293
3500 GG Utrecht Netherlands

Das **JMS** C-Mix bzw. E-Mix-System ist eine Fader- bzw. Routing-Automation, die in beliebige Mischpulte nachgerüstet werden kann. Das System besteht aus hochwertigen VCA-Baugruppen, die dann von einem ST gesteuert werden. Die Software ist komplett GEM-gesteuert und sehr komfortabel. Bis zu 64 Kanälen können mit C-Mix automatisiert werden. Am JMS-Stand gab es auch eine Rack-Version des Atari ST zu sehen: 'Studio-Computer Status SII' nennt sich das System, das zwei 1 Megabyte-Ataris mit Floppy und zwei Monitoren enthält. Vier Romports, die softwaremäßig umschaltbar sind, gehören ebenso dazu, wie eine neue Tastatur mit Trackball, die sich jeweils an einen der beiden Computer schalten läßt. Paralleler Sequencer- und Mischpult-Automationsbetrieb ist somit komfortabel möglich.

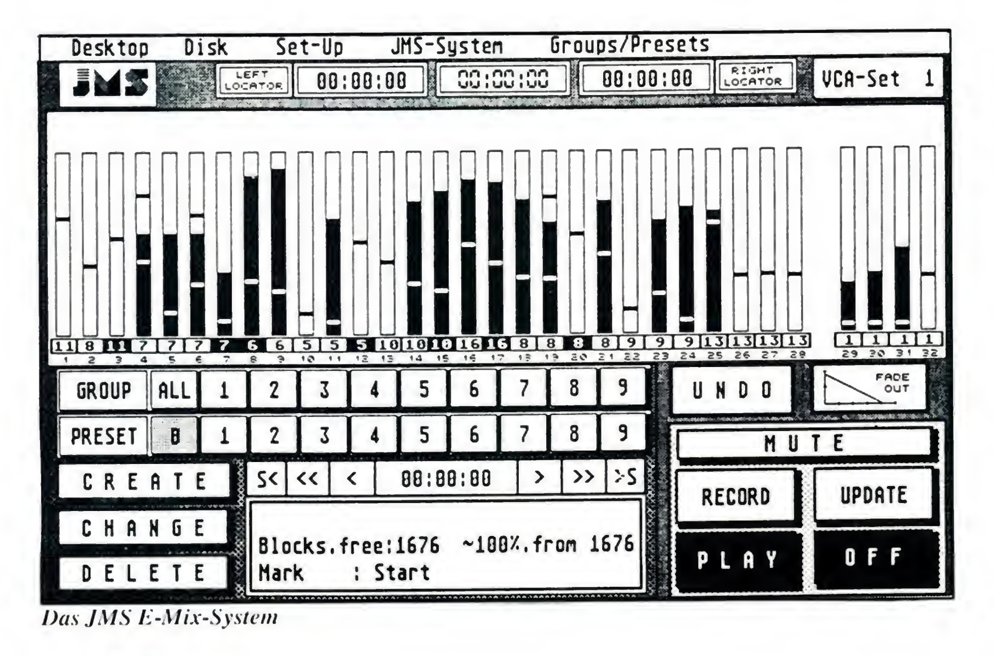
JMS
Martener Hellweg 40
4600 Dortmund 70

Status Studiobau,
Kolbstraße 10a
8034 Germering

Fader-Automationen gibt es auch von **Soundcraft** ('Twister') und J.L Cooper ('MixMate'), beide Systeme basieren auf einem Hardware-Modul mit je 8 Kanälen und optionaler Rechnerkoppelung. Mehrere Module können gekoppelt werden

Soundcraft Ltd.,
Unit 2, Borehamwood Industrial Park,
Rowley Lane,
Borehamwood, Herts WD6 5PZ

MEV
Karl-Hromadik-Str.3
8000 München 60



Das JMS E-Mix-System

Das 'Lynex'-Modul der englischen Firma **Commander** ist ein 16-Bit-Stereo-Sampler, der professionelle technische Daten aufweist: 8-stimmig bei 50 KHz Sampling-Rate, 1 Megawort eingebautes Ram, erweiterbar auf 32 Megaworte. Ein 8-kanaliger digitaler Ausgangsmixer ist integriert. Durch leistungsfähige Hardware bleibt der Rechner frei, um zum Beispiel einen Sequencer parallel laufen zu lassen. Mit dem Steinberg TwentyFour und dem C-Lab Creator wird das System in naher Zukunft kompatibel sein. Es lassen sich auch mehrere Einheiten koppeln. Ein interessanter Stereo-Sampler zu einem guten Preis.

Amptown Electroacustic GmbH
Wandsbecker Str. 26
2000 Hamburg 71

Die Musik-Mailbox des Hamburger **Music Mail Service** ist in Zukunft nicht nur über Datex P, sondern auch über ein normales Telefon-Modem zu erreichen. Die Anschlußkosten wurden auch erheblich reduziert (Basiskosten um 20.- DM monatlich), der Leistungsumfang eher erweitert: elektronische Post, Mailboxsystem, Midi-Software übers Telefon, Midi-Brett mit Sounds und Songs, sowie Konferenzmöglichkeiten mit maximal 8 Teilnehmern.

MMS
Billwerder Neuer Deich 228
2000 Hamburg 28

Schließlich noch eine Clubnachricht: In Berlin hat sich ein Midi-Verein gegründet. Eine Vereinszeitung gibt es auch. Name: Bits & Music. Die Mitgliedschaft kostet 45,- DM im Jahr.

Midi e.V. Postfach 440308
1000 Berlin 44

CS

ENDE

STARKE SOFTWARE



INTEGRIERTES SOFTWAREPAKET FÜR DEN KLEIN- UND MITTELSTÄNDISCHEN HANDEL.

Das Programm mit der hohen Geschwindigkeit und Bedienerfreundlichkeit.

Leistungsmerkmale

- Verwaltung von Artikeln, Kunden, Interessenten, Lieferanten
- Erstellung von Adress-, Inventur-, Provisions- und Postenlisten
- Datenkapazität nur von externem Speichermedium abhängig
- Erstellung von Bestellungen, Angeboten, Auftr.-Bestätigungen, Lieferscheinen, Rechnungen, und Gutschriften
- Automatisch Bestandsüberwachung
- Automatisches Mahnwesen
- Einfache Weiterverarbeitung von bereits erstellten Dokumenten
- MWSt-Änderung für alle Artikel mit einem(!) Befehl
- 4 VK-Preise pro Artikel, jeder Artikel einzeln rabattierbar
- 1 Zusatz-Preisfeld pro Artikel
- Kunden- und Artikelumsatzstatistik
- Komfortable Druckeranpassung
- Schnittstelle zu 1st Word (Plus)
- Verarbeitung von 1st Word-Texten (Plus)
- Serienbrieffunktion, Etiketten
- Druckermasken für alle Dokumente frei erstellbar, somit Anpassung an Vordrucke aller Art
- einfache Benutzerführung, hohe Bedienungsfreundlichkeit
- kompakter Programmcode (nur 80 KB), dadurch hohe Verarbeitungsgeschwindigkeit
- ständige Programmpflege & UP-Date-Service
- Handbuch

DM 899,–



DIE KOMFORTABLE BUCHFÜHRUNG FÜR FREIBERUFLER UND KLEINGEWERBETREIBENDE.

Ohne Vorkenntnisse leicht zu bedienen.

Leistungsmerkmale

- Freie Wahl des Abschlußzeitraums (Periode)
- Voreinstellung von bis zu 5 Steuersätzen
- Einrichtung von bis zu 10 laufenden Konten
- Einrichtung von bis zu 200 Einnahme/Ausgabe-Konten
- Erfassung von bis zu 2.500 Buchungen pro Periode
- Volle Eingabekontrolle auf Fehler (Löschen der letzten Buchung)
- Erstellung von Bilanzen
- Erstellung von Journalen
- Erstellung von Saldenlisten und Kontenrahmen
- Datenaufbereitung für USt.-Voranmeldung
- Ausdruck aller wichtigen Daten
- Anpassung an jeden Drucker
- Schnittstelle zu anderen Programmen
- Periodenabschluß mit automatischer Saldenübertragung in die Datei für die Folgeperiode
- Voll in GEM eingebunden
- Einsetzbar auf allen ATARI ST und MEGA ST mit S/W und Farbmonitor
- UP-DATE-SERVICE
- Handbuch

DM 149,–



ORDNUNG, ÜBERBLICK, PLANUNG – MIT DER HAUSHALTSBUCHFÜHRUNG FÜR ALLE.

Leistungsmerkmale

- Monats- und Jahresabschluß
- Benenung von bis zu 50 laufenden und 200 Kostenkonten
- Verwaltung von bis zu 50 Dauerbuchungen mit wählbarer Frequenz
- Bis zu 2.500 Buchungen pro Periode
- Volle Eingabekontrolle auf Fehler
- Erstellung von Bilanzen, Journals, Saldenlisten und Kontenrahmen
- Ausgabe von Buchungslisten mit Korrekturmöglichkeit
- Saldenbildung von Kontenblöcken
- Anpassung an jeden Drucker
- Schnittstelle zu anderen Programmen
- Voll in GEM eingebunden
- UP-DATE-SERVICE
- Handbuch

DM 129,–

alle Preise sind unverbindlich empfohlene Verkaufspreise

Heim Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

BESTELLCOUPON

an Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir

____ Stück **TKC-FAKTURA ST** z. Preis v. DM 899,–
____ Stück **TKC-EINNAHME/ÜBERSCHUSS ST** z. Preis v. DM 149,–
____ Stück **TKC HAUSHALT ST** z. Preis v. DM 129,–

zuzüglich DM 5,– Versandkosten unabhängig von der Bestellmenge.

☐ per Nachnahme ☐ Scheck liegt bei ☐ per Vorausberweisung

Name, Vorname ____________

Straße, Nr. ____________

PLZ, Ort ____________

Unterschrift ____________

Benutzen Sie auch die im ST-COMPUTER vorhandene Bestellkarte

ENDLICH DA! GFA-BASIC 3.0

Es war einmal, vor langer, langer Zeit, da wurde jedem ATARI ST-Käufer ein ziemlich langsamer BASIC-Interpreter namens "ST-BASIC" mitgeliefert. Mit extrem wenigen Befehlen konnte der Aufsteiger seine Programmierkenntnisse nutzen, die er bereits anderweitig erworben hatte, denn mit dem (anfangs noch englischen) Handbuch war dies kaum möglich. Abgesehen davon lagen die Geschwindigkeiten, die die Programme erreichten, noch unter denen eines Commodore 64.

```
/|\| Save |Save,A| Quit | New  |BlkSta|Replac| Pg ⇧ |Txt 16|Direct| Run | ?4:25:55
   | Load |Merge |Llist |Block |BlkEnd| Find | Pg ⇩ |Insert| Flip | Test|       20
PRINT AT(15,4);"Bitte geben Sie eine Zahl zwischen 1 und 8 ein"
PRINT AT(15,6);
INPUT z
t$="Die eingegebene Zahl ist "
auswertung(t$)       ! Wie GOSUB/@ auswertung
PRINT AT(15,9);t$
END
'
PROCEDURE auswertung(VAR text$)
  SELECT z
  CASE 1
    text$=text$+"1"
  CASE 2 TO 5
    text$=text$+"zwischen 2 und 5"
  CASE 6,7,8
    text$=text$+"6,7 oder 8"
  DEFAULT
    text$=text$+"nicht zwischen 1 und 8"
  ENDSELECT
RETURN
```

Bild 1: Ein einfaches Beispiehl für eine Mehrfachverzweigung mit SELECT und CASE.

Lang ersehnt erschien gegen Juli 1986 endlich ein neues BASIC, das GFA-BASIC V1.0. Überglücklich wagte man sich ans Programmieren, was nach Umstellung auf die fehlenden Zeilennummern zu einem richtigen Vergnügen ausartete. Nicht nur die vielen zusätzlichen Funktionen, auch die Geschwindigkeit begeisterte. Etwas später wurde es erweitert zum GFA-BASIC V2.0. Wieder zollten die BASIC-Programmierer GFA hohes Lob, da die Version V2.0 alles bisher Dagewesene übertraf. Dann erschien das OMIKRON-BASIC, was noch besser, noch schneller, aber in der Bedienung doch ganz anders war.
Aber nun schlug GFA-Systemtechnik in Düsseldorf zurück !
Am 30. März 1988 wurde uns anläßlich einer Pressekonferenz die Vorabversion des neuen GFA-BASICs, Version 3.0, übergeben. Wir hatten Gelegenheit, die Vorabversion des neuen Super-BASICs zu testen und waren mehr als überrascht, was in dem neuen GFA-BASIC V3.0, das nicht einmal 90 KByte umfaßt, alles drinsteckt.
Dieses BASIC übertrifft mit Abstand alle bisher auf dem ATARI ST bekannten BASIC-Dialekte in Bezug auf die Befehls- und Funktionsvielfalt. Hatte die Version V1.0 schon rund 180 Befehle und Funktionen, so brachte es die Version V2.0 auf immerhin etwa 210. Die neue Version V3.0 wirft ca. 390 Befehle in die Waagschale, wobei hier natürlich die neu eingebundenen AES- und Line-A-Befehle den Löwenanteil stellen.
Da immer noch der Streit zwischen den Programmierfreaks herrscht, ob BASIC als Programmiersprache überhaupt ernst zu nehmen sei, wurde mit GFA 3.0 nun ihr Ansehen weiter gesteigert. Schließlich ist BASIC im Home- und Personal-Computerbereich noch immer eine der am weitesten verbreiteten Programmiersprachen überhaupt, nicht nur wegen der äußerst leichten und schnellen Erlernbarkeit.
Wir wollen hier jedoch nur die Erweiterungen und Veränderungen gegenüber der Version 2.0 erwähnen, da sonst der Umfang des ganzen Heftes gesprengt würde.

Der Editor

"Schneller, höher, weiter", das war schon das Prinzip der alten Römer. Dies hat auch GFA erkannt und den Editor noch weiter verbessert. Gegenüber der alten Version wurde vieles verändert und erweitert.
Am auffälligsten ist sicherlich die Uhrzeit, die ständig angezeigt wird. Hier kann sich der Programmierer freuen, der um 22 Uhr ins Bett wollte, und um 5 Uhr morgens bemerkt, daß es "ja schon so spät" ist. Abgesehen davon wird ständig die aktuelle Programmzeile angezeigt, in der sich der Cursor gerade befindet. Dies ist vor

allem dann extrem nützlich, wenn man mit Hilfe einer Cross-Referenzliste versucht, irgendwelche Fehler zu finden und zu entfernen. Ferner ist es möglich, direkt die Nummer der Programmzeile einzugeben, in die man springen will, was prompt vom Interpreter erledigt wird.
Nun kann man außerdem zwischen mehreren Programmzeilen wechseln, ohne daß sich der Cursor dauernd an den Anfang der Zeile bewegt. Um dies noch zu erleichtern, läßt sich der Ziffernblock der Tastatur als Cursortastenblock umbelegen. Hier hat man dann die Möglichkeit, ähnlich einem PC, seinen Cursor um ein Zeichen in alle Richtungen zu bewegen, eine Seite vor- oder zurückzublättern, beziehungsweise gleich ans Ende oder den Anfang des Programmes zu springen. Weiterhin wurden sehr viele Tasten in Kombination mit der CONTROL-Taste überaus nützlich belegt, was viel Arbeit spart.
Ein weiterer riesiger Vorteil besteht darin, daß sich die Funktionstasten endlich mit Texten belegen lassen. Über SHIFT und ALTERNATE stehen insgesamt 20 Funktionstasten zur beliebigen Belegung mit Befehlen, Texten oder anderem zur Verfügung. Die alten Funktionstastenbelegungen bleiben jedoch weiterhin erhalten. Weiter verbessert wurde auch die Such- und Ersatztexteingabe. Hatte man sich früher einmal vertippt, so mußte man den gesamten Suchtext erneut eingeben. Jetzt wird der alte Text angezeigt und kann jederzeit nach Belieben verändert werden.
Eine weitere Augenfreude für den Programmierer ist die Möglichkeit, seine Prozeduren auf- oder zuzu"klappen". Man kann durch kurzen Tastendruck eine ganze Unterprozedur auf eine Zeile verkürzen. Dies schafft mehr Übersichtlichkeit bei längeren Programmen. Somit lassen sich im Listing äußerst übersichtliche Programme darstellen, die scheinbar nur aus sehr wenigen Zeilen bestehen.

```
/|\| Save |Save,A| Quit | New  |BlkSta|Replac| Pg ⇧ |Txt 16|Direct| Run | 14:26: :8
   | Load |Merge |Llist |Block |BlkEnd| Find | Pg ⇩ |Insert| Flip | Test|        :8
baum0%=0   ! TREE #0
datum%=4   ! Objekt in TREE #0
'
fehler%=RSRC_LOAD("G:\DATUM.RSC")          ! RSC einladen
'
fehler%=RSRC_GADDR(0,baum0%,tree_%)        ! Adresse des Baumes nach tree_% holen
'
' (Das Zeichen ~ entspricht dem Befehl VOID)
'
~FORM_CENTER(tree_%,nx%,ny%,nw%,nh%)      ! Dialogbox zentrieren lassen
~OBJC_DRAW(tree_%,0,8,nx%,ny%,nw%,nh%)    !  -  "  -  zeichnen
~FORM_DO(tree_%,datum%)                    !  -  "  -  aufrufen
'
dat$=CHAR{{OB_SPEC(tree_%,datum%)}}        ! Eingegebenen Text (Datum) auslesen
PRINT LEFT$(dat$,2)+"."+MID$(dat$,3,2)+"."+RIGHT$(dat$,4)     ! und ausgeben
'
~RSRC_FREE()                               ! Speicher des RSC wieder freigeben
END
```

Bild 2: Auch die AES-Routinen sind jetzt implementiert.

Noch mehr Datentypen

Bei der Version 2.0 gab es die "normalen" Integerzahlen, die 32 Bit (4 Byte) lang waren, die Fließkommazahlen mit 48 Bit (6 Byte), die Stringvariablen mit einer Maximallänge von 32767 Zeichen und die Boolean-Variablen, die die Werte -1 (TRUE) und 0 (FALSE) annehmen konnten und nur 1 Bit umfaßten (in Arrays, sonst 1 Byte).
Bei der neuen Version existieren die Typen BOOLEAN (1 Bit), BYTE (1 Byte lang), CARD (2 Byte, ohne Vorzeichen), INT (2 Byte, mit Vorzeichen), LONGINTEGER (4 Byte), SINGLE (4 Byte, IEEE-Format - einfache Genauigkeit), DOUBLE (8 Byte, IEEE-Format - doppelte Genauigkeit), FLOAT (8 Byte, GFA-3.0 Format) und CHAR (Zeichenkette mit Nullbyte). Gekennzeichnet werden die einzelnen Variablen ebenso wie im OMIKRON BASIC mit sogenannten Postfixen, Dies sind Zeichen, die der Variablen hintenangestellt werden, zum Beispiel a% für die Integer-Variable a usw.
Außerdem können den Variablen, die mit einem bestimmten definierbaren Buchstaben beginnen, feste Datentypen zugewiesen werden, zum Beispiel alle Variablen mit b sind integer, alle mit c float usw.
Die Arithmetikgenauigkeit wurde von 11 Stellen auf 13 Stellen erhöht. Die Trigonometrischen Funktionen wurden um ASIN, ACOS (Arcussinus, Arcustangens), SINQ und COSQ (interpolierter Sinus und Cosinus) ergänzt.

Bitoperationen

Um die Geschwindigkeit der Programme weiter erhöhen zu können, wurden diesesmal auch maschinennahe Befehle implementiert, die die Beeinflussung von einzelnen Bits ermöglichen und somit das lästige ODERn und UNDen endlich in Vergessenheit geraten lassen. Es lassen sich einzelne Bits löschen (BCLR), setzen (BSET), abfragen (BTST) und auch negieren (= exklusiv verodern, BCHG). In Anlehnung an die Maschinensprache wurden auch Bit-Verschiebebefehle installiert. Hiermit kann man Bytes, Worte oder Langworte um eine beliebige Anzahl an Bits nach links (SHL) oder rechts (SHR) verschieben. Außerdem lassen sich die Bits auch rotieren (ROL, ROR); d.h. daß die "rausgeschobenen" auf der anderen Seite wieder "hereingeschoben" werden. Und für diejenigen, die oft mit Langworten jonglieren müssen, gibt es die Möglichkeit, LO- und HI-Wort zu vertauschen (SWAP()). Ferner lassen sich auch Zahlenwerte ganz einfach mit AND(), OR(), XOR(), IMP() und EQV() manipulieren. Genauso kann man aus mehreren Bits (zum Beispiel aus einem Langwort mit 32 Bit) einen

Teil auslesen, zum Beispiel 8 Bit (BYTE(), CARD(), WORD()).

Strukturbefehle

Die Essenz eines jeden Programms besteht meist aus den hunderten von Vergleichen. IF ... THEN ... ELSE ... ENDIF. Bisher war man auf diese vier Befehle beschränkt. Doch jetzt wird das Programmiererherz wieder höher schlagen. Zur Verfügung stehen auch ELSEIF, SELECT, CASE TO, DEFAULT, OTHERWISE, ENDSELECT und CONT. Damit ist endlich eine gezielte Mehrfachabfrage möglich. Man muß nicht mehr viele IFs hintereinanderschachteln, sondern kann mittels CASE ganz einfach jeden einzelnen Fall auswerten.
Um Umsteigern von anderen Programmiersprachen den Weg zu erleichtern, wurden die Befehle wie NEXT, UNTIL, WEND, LOOP um zusätzliche erweitert: ENDFOR, ENDREPEAT, ENDWHILE, ENDDO. Die einfache DO ... LOOP-Endlosschleife wurde wesentlich verbessert. Neu sind DO WHILE, DO UNTIL, LOOP WHILE, und LOOP UNTIL, die eine noch effektivere Schleifenprogrammierung ermöglichen. Auch bei den Unterprogrammen ist man nicht mehr stur gezwungen, RETURN ans Ende zu setzen. ENDPROC und ENDSUB stehen zusätzlich zur Verfügung.

Mehrzeilige Funktionsprogrammierung

Ließen sich früher Funktionen mittels DEFFN immer nur einzeilig entwerfen, so gibt es nun die Möglichkeit, sie mehrzeilig wie eine Prozedur zu definieren (FUNCTION, ENDFUNC). Außerdem läßt sich am Schluß ein ganz bestimmter Variablenwert zurückgeben (RETURN x). Für einige Anwendungszwecke ist es nötig, Variablen direkt zu manipulieren und am Ende eines Unterprogrammes den neuen Wert an die alte Variable zurückzugeben. Bei Version 2.0 mußte man im Unterprogramm entweder den gleichen Variablennamen benutzen, was spätestens beim zweiten Aufruf zu Kollisionen führte und äußerst unflexibel war. Oder man mußte über Pointer die Variable indirekt verändern, was jedoch meist etwas umständlich verlief. Doch auch diesem kleinen Manko wurde abgeholfen. Ab sofort ist es möglich, eine Variable nicht als Wert zu übergeben, sondern als Variable selbst (VAR). Somit kann man den Variableninhalt beliebig oft verändern, ohne den gleichen Variablennamen benutzen zu müssen. Aber dennoch ist es möglich, ein solches Unterprogramm individuell zu nutzen. Diese Möglichkeit wird in erster Linie auch denjenigen große Erleichterung bringen, die gerne mit rekursiven Prozeduren - also Unterprogrammen, die sich selbst aufrufen - arbeiten.

GFA-BASIC
Save
Load
Deflist
Defint

Dies ist eine normale Dialogbox
ST Computer 1988, Test GFA-BASIC V3.0

Bitte geben Sie das Datum ein

Datum :07/04/1988

Ende

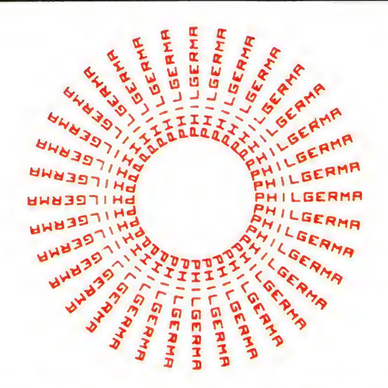
Bild 3: Die Menüleiste des GFA-BASICs mit dem Ergebnis des kleinen Programms aus Bild 2.

Interruptprogrammierung

Bei OMIKRON hat man wohl die Möglichkeit der Interruptprogrammierung abgeschaut, hat diese jedoch wesentlich eleganter gelöst. Es ist nicht nur der stupide Aufruf eines Unterprogrammes nach einer bestimmten Zeit möglich, der Aufruf kann einmalig oder unendlich oft geschehen. Die Zeit kann in tausendstel Sekunden angegeben werden, wobei jedoch höchstens alle zwei Millisekunden ein Aufruf stattfinden kann. Hierzu dienen die Befehle EVERY und AFTER. Die Anwendungsmöglichkeiten der Interruptprogrammierung sind äußerst weit gestreut, von der nebenbei ablaufenden SOUND-Demonstration, über Bildschirmschwarzschaltung nach bestimmter Zeit, Passwortabfragen bis hin zu Bildschirmspielereien oder eventuell sogar periodisch auftauchenden Viren (hoffentlich gutartige).

Fehlersuche

Aber es wurde auch an die genervten Programmierer gedacht, die nach stundenlanger Fehlersuche oft beabsichtigten den Computer aus dem Fenster zu werfen, weil der Programmfehler immer noch nicht entdeckt war. Waren sie früher noch auf TRON und TROFF angewiesen, so können sei heute mittels TRON PROCEDURE nach jeder Befehlsausführung auch in eine TRACE-Prozedur hineinspringen und dort den letzten Befehl, der in der Variable TRACE$ zu finden ist, auswerten. Auch muß sich der Anwender nicht mehr mühsam alle Felder eines Arrays per Hand im Direktmodus ausprinten lassen. Mit dem Befehl DUMP lassen sich alle Variableninhalte auf den Bildschirm oder in eine Datei ausgeben.



PHILGERMA IHR SPEZIALIST FÜR ATARI COMPUTERSPRACHEN

LATTICE C (Metacomco) – Neueste Version 3.04 des bewährten Standard-Compilers der IBM-Welt. Voller Kernigham/Ritchie-Standard. Floating-Point-Arithmetik mit 16 Stellen Genauigkeit. Natürlich die VDI/AES Funktionen, ein sehr guter Resource-Construction-Editor. Utilities (Symbolischer Debugger, MAKE, Shell, Disassemb.) und über 320 UNIX-ähnliche Routinen. Dieses Entwicklungspaket wird mit 600 Seiten starkem deutschen Handbuch geliefert .... DM 298,00

MCC PASCAL2 (Metacomco) – Pascal Compiler ISO 7185 Standard. Schneller 1-Pass Compiler. 64-Bit IEEE Fließpunktarithmetik. Natürlich die VDI/AES Funktionen, ein sehr gutes Resource-Construction-Editor. Utilities (Symbolischer Debugger, MAKE, Shell, Disassemb.) MCC Pascal2 Programme können mit anderen Sprachen gelinkt werden.
Deutsches Handbuch 600 Seiten ...... DM 248,00

MCC ASSEMBLER (Metacomco) – Professioneller Makro Assembler, der den vollen Motorola 68000 Instruktion Set unterstützt. Mit Editor, Linker und TOS-Libraries, AES, VDI. Ausführliche Fehlermeldungen.
Deutsches Handbuch .... DM 168,00

CAMBRIDGE LISP (Metacomco) – Interpreter und Compiler mit dem Sprachumfang, den man von Großrechnern gewöhnt ist. Volle REAL-Arithmetik 16 MByte Adressraum. Handbuch 330 S. ...... DM 448,00

BCPL Interpreter (Metacomco) – Eine leistungsfähige Sprache, besonders für die Systemsoftwareentwicklung. Mit GEM-Bibliothek .... DM 298,00

MODULA-2 (TDI) – Diese umfangreiche Modula Implementierung vereinigt die Vorteile von Pascal mit neuen Sprachelementen. Integriertes Programmierentwicklungssystem. Compiler, Editor, Debugger und GEM Einbindung. V 3.01 .... DM 248,00

MODULA 2 Developer zusätzlich den Resource-Construction Editor, RAM-Disk, Spooler und umfangreicher Bibliothek V 3.01 .... DM 398,00

MODULA 2 Commercial – zusätzlich alle Utilities im Quellkode V 3.01 .... DM 698,00

OMIKRON BASIC (Omikron) – Ein sehr schneller 'Halbcompiler', der die strukturierte Programmierung unterstützt. Rechengenauig. 19 Stellen 99%-ig kompatibel zu M-Basic, kompl. GEM-Library, alle AES/VDI-Funktionen verfügbar, Extras wie Masken-Input, Sort- und Matrizenbefehle, mit 180-seitigem deutschen Handbuch .... DM 178,00
OMIKRON BASIC Compiler .... DM 178,00

PRO PASCAL (Prospero) V2.11 – Professioneller Compiler (ANSI/ISO/BSI-Standard). Gegenüber ISO viele Erweiterungen. Einbindung von Assembler und Fortran 77 .... DM 248,00
PRO FORTRAN77 (Prospero) – ANSI X3.9-78 Standard Fortran 77-Compiler. Optimierender 2-Pass-Compiler mit vielen zus. Optionen. 68881-Coprozessor direkt ansprechbar. Ein Fortran wie auf Großrechnern .... DM 378,00
AC FORTRAN 77 absoft .... DM 448,00
TRUE BASIC Interpreter .... DM 248,00
TRUE BASIC Runtime Package ...... DM 248,00
TRUE BASIC 3D-Grafik Packet .... DM 128,00

NATÜRLICH HABEN WIR AUCH SPIELE:
Bard's Tale 89,– DM; Defender of the Crown 89,– DM; Dungeon Master 79,– DM; Frost Byte 59,– DM; Guild of Thieves 59,– DM; Gunship 79,– DM; Impact 39,– DM; Indoor Sports 79,– DM; Jinxter 69,– DM; Moebius 79,– DM; Star Wars 59,– DM; Test Drive 99,– DM; Western Games 59,– DM; Wizball 49,– DM – und viele mehr!
– Alle Infocom Text-Adventures vorrätig –

GFA BASIC Interpreter .... DM 98,00
GFA BASIC Compiler .... DM 98,00
GFA BASIC 68881 Interpreter/Comp. .. DM 348,00
LDW BASIC Compiler Atari Basic komp. DM 98,00

Star Writer ST Textverarbeitung ...... DM 198,00
1st WORD PLUS deutsch .... DM 198,00
1st Proportional Utility .... DM 88,00
SIGNUM 2 Text- und Grafik-Programm . DM 428,00
Timeworks Publisher DTP .... DM 388,00

TIM V1.1 Ein universelles Buchführungsprogramm mit Geld- und Gegenkonten .... DM 298,00
TIM Cashflow Kassenbuchprogramm als Ergänzung zu TIM. Mandantenfähig .... DM 298,00
K-SPREAD 2 Tabellenkalkulation ...... DM 228,00
K-GRAPH 2 Grafik + Statistik .... DM 148,00
K-COMM 2 Terminalprogramm VT100 .. DM 148,00

CADprojekt – Konstruktionsprogramm
Testberichte, die überzeugen!
Normalversion .... DM 298,00
Vollversion mit Plottertreiber
und vollautom. Bemaßung .... DM 798,00
Demoversion o. Ausgabem. .... **DM 15,00**

AUSZUG AUS UNSERER HARDWARELISTE:
Einzellaufwerk 3,5" 720KB .... DM 348,00
Doppellaufwerk 3,5" 2★720KB ...... DM 648,00
Einzellaufwerk 5,25" 40/80 Spuren ... DM 448,00
Speichererweiterung auf 2,5 MByte ... DM 1098,00
10 Disketten 3,5" 2DD no name ..... DM 29,00

HARDDISK XEBEC Alle Geräte im Metallgehäuse m. Platz für 2 Laufwerke, 8 Partitions möglich
Harddisk 40 MB 28mS komplett ..... DM 2598,00
Harddisk 70 MB 28mS komplett ..... DM 3998,00
Harddisk 40 MB mit Tape-Streamer ... DM 4398,00

K-MAX Transputerboard .... DM 2598,00

Cascade OCCAM Transputer-Software DM 598,00
OMEGA Farbgrafikkarte 1024★512 ... DM 2698,00

PHILGERMA PRÄSENTIERT

BASiCALC

- einfache Bedienung unter GEM; Online-Hilfe aufrufbar.
- Programm und Handbuch komplett in deutsch.
- über 130 mathematische und finanztechnische Funktionen ähnlich Lotus 1-2-3
- bis zu 10 000 Zeilen oder 676 Spalten nach vorhandenem Speicherplatz.
- Datenaustausch mit anderen Programmen über ASCII- oder DIF-Dateien.
- läuft auf jedem ATARI ST in Farbe oder s/w; einfache Druckeranpassung.

78,– DM

PC-DITTO

Software-MS-DOS-Emulator

Version 3.64
(vorher bezogene Versionen daten wir up!)
volle Unterstützung der deutschen Tastatur;
für Farb- und s/w-Monitor
Hohe Kompatibilität für nur

198,– DM

Die GEM Produktfamilie jetzt auch auf dem ATARI ST verfügbar!

- **GEM** Diary (incl. GEM System u. GEM Desktop) 169,– DM
- **GEM** Write + **GEM** Paint .... 399,– DM
- **GEM** Graph .... 369,– DM
- **GEM** Draw Plus .... 369,– DM
- **GEM** Wordchart .... 369,– DM

Dazu wird benötigt:
GEM SYSTEM SOFTWARE V. 2.2
und 1 MB RAM-Speicher

GEM System 2.2 wird geliefert zus. mit GEM Diary.

DÜRER

Nicht nur ein Zeichenprogramm

- komfortable Bedienung
- alle wichtigen Zeichenfunktionen
- Sprühdose
- Textfunktionen
- Lassofunktion, 2 Lupenmodi
- volle Bildanimation
- Diashow mit reizvollen Überblendeffekten
- Bilder mit Signum2 weiterzuverarbeiten
- DÜRER IST MIT GFA-BASIC PROGRAMMIERBAR!

Für 78,– DM

OS-9

von CUMANA
auf dem ATARI ST implementiert
mitgeliefert: ROM-Modul
Compiler: C.Pascal, Basic, Assembler, Textverarbeitung
Datenbank, Tabellenkalkulation, Echtzeituhr, Handbücher

alles zus. **1598,– DM**

Fordern Sie unsere umfangreiche Preisliste mit Produktbeschreibungen an! Oder lassen Sie sich in unserem Laden alle Produkte vorführen oder testen Sie sie selbst.
Händler bitte Händliste anfordern

Tel. Bestellannahme Mo.–Fr. 10–18 h, Sa. 10–14 h. (Tel. 0 89/28 12 28)

4,80 DM Versandkostenanteil bei Bestellungen unter 200,– DM.

Nachnahmegebühr 3,80 DM.

Lieferung ins Ausland nur gegen Vorkasse (Euroscheck oder Postüberweisung)

PHILGERMA
GmbH & Co. KG
Barerstr. 32
8000 München 2
Tel. 0 89/28 12 28



Grafikprogrammierung

Sicherlich kennt man von Malprogrammen (wie zum Beispiel STAD) die Möglichkeit, einen ausgefüllten Kreis mit oder ohne Begrenzungslinie zeichnen zu lassen. Mußte man diesen Parameter früher mühsam über die VDI-Routinen einstellen, so genügt nun ein einziger Befehl, die Begrenzung ein oder auszuschalten: BOUNDARY.

Auch bei dem Befehl FILL hat man sich in Düsseldorf etwas einfallen lassen, was bereits bei besseren Grafikprogrammen Anwendung gefunden hat. War eine Fläche einmal nicht bis auf den letzten Punkt begrenzt, so konnte es schon einmal vorkommen, daß der Füllvorgang "übergelaufen" ist und die ganze Zeichnung zerstört war. Nun kann man auch so füllen, daß die Füllfarbe nicht mehr zu weit fließt, sondern bei Berühren bestimmter Punktfarben oder Begrenzungen mit kleiner Lücke stoppt.

Aber auch die Fans der Programmiersprache LOGO sollten nicht zu kurz kommen. Sicherlich habe sich einige an die Befehle FORWARD 10, RIGHT 90, BACK 20, LEFT 45 usw. gewöhnt. Im GFA-BASIC 3.0 können nen ab sofort die gleichen Befehle verwendet werden. Zur Verfügung stehen FD (Forward), BK (Backward), SX und SY (Scale x/y), LT (Left) und RT (Right) zur Bewegung des Zeichenstiftes und zur Drehung in eine bestimmte Richtung. Die Richtung kann auch gradweise absolut bestimmt werden mit TT (Turn To). Mit MA (Move Absolut) läßt sich der Stift an eine bestimmte Koordinate setzen bzw. kann mit DA (Draw Absolute) von der aktuellen Position an die neue eine Linie gezeichnet werden. Mit MR (Move Relative) und DR (Draw Relative) kann man den letzten Punkt als Bezugsnullpunkt verwenden. Mit CO (Color) läßt sich die Zeichenfarbe des Stiftes bestimmen. Ebenso wie in LOGO kann auch der Zeichenstift vom Papier abgehoben (PU - Pen Up) oder wieder auf das Papier heruntergelassen (PD - Pen Down) werden. Mit den sechs Variablen DRAW(0) bis DRAW(5) lassen sich die X- und die Y-Position abfragen. Außerdem können der Wert des momentan eingestellten Winkels des Zeichenstiftes, die Skalierung in X- und in Y-Richtung und der Zustand des Stiftes (-1=PD, 0=PU) ermittelt werden.

Die Besitzer eines MEGA ST's können aufatmen. Der Programmierer des GFA-BASICs 3.0 hat auch daran gedacht, die Maus an eine bestimmte Position setzen zu können. Diese Möglichkeit wird ja bereits bei manchen Programmen erfolgreich angewandt. Den Trick der Mauskoordinatenmanipulation ist in Ausgabe 4 (April) in der ST-Ecke auf Seite 143 genauer beschrieben. Da dieser jedoch vom neuen TOS abhängig ist, kann man mittels des Befehls SETMOUSE x,y nicht auf allen ST's die Mauskoordinaten einstellen.

LINE-A Routinen nun implementiert

Und wieder ging ein Traum in Erfüllung. GFA-Systemtechnik hat sich hingesetzt und die LINE-A Routinen gleich als Befehle eingearbeitet.

Dies bringt in erster Linie einen Geschwindigkeitsvorteil. Nichtsdestotrotz blieben alte und bewährte Befehle, wie zum Beispiel PLOT oder LINE erhalten. Für die LINE-A Benutzer stehen jedoch insgesamt neun Routinen zur Verfügung.

Hierzu gehört ACLIP. Mit diesem Befehl läßt sich am Bildschirm ein Begrenzungsrechteck festlegen und ein- bzw. ausschalten. Alle Grafikausgaben, die dann über diesen Bereich hinaus erfolgen, werden abgeschnitten. Die Funktion wird also in etwa der eines Fensters gerecht. Mittels dem Befehls PSET lassen sich Punkte am Bildschirm in einer gewünschten Farbe setzen. Der Befehl entspricht also in etwa PLOT. Mit PTST kann die Farbe eines Punktes ermittelt werden. Hierzu wird die Farbe aus dem Speicher ausgelesen und übergeben (wie bei POINT()). Mit dem Befehl ALINE kann man eine Linie zwischen zwei Koordinaten zeichnen. Außerdem kann die Punktfarbe vorgegeben und ein Linienmuster festgelegt werden. Zusätzlich wird noch der Grafikmodus eingestellt (Replace, Transparent, Invertiert oder Revers). Mit HLINE kann man horizontale Linien zwischen zwei Punkten auf gleicher Höhe zeichnen lassen. Diese Funktion ist noch schneller als ALINE. Auch hier lassen sich Linienmuster, Graphikmodus und Farbe angeben. Mit ARECT kann man ein gefülltes Rechteck mit eckigen Ecken auf den Bildschirm bringen. Hier lassen sich Füllfarbe und ein beliebiges Füllmuster (bitweise) angeben (entspricht in etwa PBOX).

Fensterverwaltung

Bei der Fensterverwaltung kam nicht allzuviel hinzu. W_HAND liefert die Handle Nummer eines GEM-Fensters. Mit W_INDEX kann man die Fensternummer an ein GEM-Handle zurückmelden. Der Befehl TOPW (Topwindow) wurde eingeführt, um dem Benutzer das aufwendige Anklicken eines der hintenliegenden Fenster zu ersparen. Ruft man ihn auf, so wird das Fenster mit der entsprechenden Nummer aktiv und kommt wieder in den Vordergrund.

VDI-Aufrufe

Es wurde auch an die GEM-Routinen gedacht. Um den Umgang mit dem VDI nicht weiterhin mühsam über DPOKEs einstellen zu müssen, kann ein Teil der CONTRL, INTIN und PTSIN Variablen direkt über den Befehlsaufruf vorgegeben werden. Außerdem läßt sich auch der Subopcode, also die Unterroutine einer VDI-Routine mit dem Befehl VDISYS spezifieren.

Dazu wurde ein WORK_OUT() Array installiert, mit dem sich alle nötigen Parameter abfragen lassen. Dies kann die Bildschirmbreite, -höhe, die Anzahl der verfügbaren Farben und vieles andere eines angeschlossenen Gerätes (zum Beispiel Monitor) sein.

GEM-AES, Benutzung ganz einfach

Wieder wurde an die Assemblerprogrammierer gedacht. Die einzelnen Datenregister D0 bis D7 lassen sich genauso wie die Adressregister A0 bis A6 mittels RCALL mit beliebigen Werten vorbelegen. Somit hat das mühsame gePOKEe um Werte an Maschinenspracheprogramme zu übergeben endlich ein Ende. Natürlich können über dieses Array auch Werte aus den Datenregistern ausgelesen werden, so daß ein reger Datenaustausch möglich ist.

Die Routinen des AES wurden reichlich implementiert. Zu finden sind verschiedene Bibliotheken. Zum einen die applications library. In ihr finden sich die Befehle, die mit Applikationen zu tun haben. Dies sind APPL_INIT, APPL_EXIT, die bei Start oder Ende des GFA-Programmes automatisch ausgeführt werden. Mit APPL_READ lassen sich Bytes aus dem Ereignisbuffer lesen, mit APPL_WRITE in den Buffer schreiben. Durch APPL_FIND läßt sich eine Anwendung finden und die Identifikationsnummer kann ermittelt werden. Die Befehle APPL_TPLAY und APPL_TRECORD zum Abspielen und Aufzeichnen bestimmter Ereignisse vervollständigen die Reihe der Applikationsbefehle.

Aber es wurde auch die event library implementiert. Mit Hilfe von EVNT_KEYBD kann auf ein Tastaturereignis gewartet werden. Mit EVNT_BUTTON kann man auf Mausklicks warten. Durch den Befehl EVNT_MOUSE kann man darauf warten, daß die Maus einen rechteckigen Bildschirmbereich verläßt oder betritt. Dies kann zum Beispiel dazu verwendet werden, um bei einem Grafikprogramm die Maus im Zeichenfeld als Pinsel erscheinen zu lassen usw. Durch EVNT_MESAG kann eine Mitteilung in den Ereignisbuffer geschrieben werden. EVNT_TIMER wartet eine angegebene Zeit von Millisekunden. Der wohl interessanteste Befehl für events ist EVNT_MULTI. Mit ihm lassen sich die verschiedensten Ereig-

Übersicht über die Befehle und Funktionen des neuen GFABASIC 3.0

DEFxxx
DEFLIST

Felder
DIM
DIM?
OPTION BASE
ARRAYFILL

Typumwandlung
ASC()
CHR$()
STR$()
BIN$()
OCT$()
HEX$()
VAL()
VAL?()
CVx()
MKx$()
CINT()
CFLOAT()

Pointeroperationen
*
BYTE{}
CARD{}
INT{}
LONG{}
{}
FLOAT
SINGLE{}
DOUBLE{}
CHAR{}
xPEEK
xPOKE
V:
VARPTR
ARRPTR
ABSOLUTE

Löschen und Vertauschen
CLEAR
CLR
ERASE
SWAP
SSORT
QSORT
INSERT
DELETE

Reservierte Variablen
FALSE
TRUE
PI
DATE$
TIME$
TIMER
SETTIME

Spezielles
LET
VOID
~

Speicherverwaltung
FRE
BMOVE
BASEPAGE
HIMEM
RESERVE
MALLOC
MSHRINK
MFREE

Mathematische Funktionen
ABS
SGN
ODD
EVEN
INT
TRUNC
FIX
FRAC
ROUND
MAX
MIN
SQR
EXP
LOG
LOG10
SIN
COS
SINQ
COSQ
TAN
ATN

Zufallszahlengenerator
RND
RANDOM
RAND
RANDOMIZE

Befehle und Arithmetik
DEC
INC
ADD
SUB
MUL
DIV
PRED()
SUCC()
ADD()
SUB()
MUL()
DIV()
MOD()

BIT-Operationen
BCLR
BSET
BTST
BCHG
SHL
SHR
ROL
ROR
AND()
OR()
XOR()
IMP()
EQV()
SWAP()
BYTE()
CARD()

Zeichenkettenverwaltung
LEFT$
RIGHT$
MID$ (als Befehl und Funktion)
PRED
SUCC
LEN
INSTR
RINSTR
STRING$
SPACE$
SPC
UPPER$
LSET
RSET

Tastatureingabe
INKEY$
INPUT
LINE INPUT
FORM INPUT
FORM INPUT AS

Bildschirmausgabe
PRINT
PRINT AT()
LOCATE
PRINT USING
PRINT AT() USING
DEFNUM
CRSCOL
CRSLIN
POS
TAB
HTAB
VTAB
KEYPAD
KEYTEST
KEYGET
KEYLOOK
KEYPRESS
KEYDEF

Liste geht weiter...

nisse kontrollieren. Zu guter letzt kann mit EVNT_DCLICK die Geschwindigkeit für Doppelklicks der Maus eingestellt werden.

Zur Programmierung von Pull-Down-Menüs gibt es die Befehle MENU_BAR zum Zeichnen eines Menüs, MENU_ICHECK um Häkchen vor einem Eintrag zu setzen oder zu löschen, mit MENU_IENABLE lassen sie sich als anwählbar oder nicht anwählbar darstellen. MENU_TNORMAL stellt einen Eintrag normal oder invers dar. MENU_REGISTER gibt die Namen der maximal sechs Accessories aus.

Zur Manipulation und Darstellung von Dialogboxen und ähnlichem gibt es die object library, mit den Befehlen OBJC_ADD, OBJC_DELETE, OBJC_DRAW um ein Objekt zu zeichnen, OBJC_FIND zum finden, OBJC_ORDER um die Objektreihenfolge zu beeinflussen, OBJC_OFFSET zu Koordinatenberechnung, OBJC_EDIT und OBJC_CHANGE.

Speziell zu Kontrolle dient die form library mit FORM_DO, FORM_DIAL, FORM_ALERT, FORM_ERROR, FORM_KEYBD und FORM_BUTTON.

Die graphics library mit GRAF_RUBBERBOX ist zum Zeichnen bestimmter Markierungen, zum Beispiel ein sich bewegendes Rechteck, gedacht. Weiterhin stehen GRAF_DRAGBOX, GRAF_MOVEBOX, GRAF_GROWBOX, GRAF_SHRINKBOX, GRAF_WATCHBOX, GRAF_SLIDEBOX und GRAF_HANDLE zur Verfügung, deren Name meist schon auf ihre Funktion hindeutet. GRAF_MOUSE und GRAF_MKSTATE runden die Befehlsvielfalt ab.

Durch SCRP_READ und SCRP_WRITE ist der Datenaustausch zwischen verschiedenen Applikationen möglich. Auch die Fensterverwaltung wurde wesentlich vereinfacht. Mit WIND_CREATE, WIND_OPEN, WIND_CLOSE, WIND_DELETE, WIND_GET, WIND_SET sind dem "fensterln" ebenso wenig Grenzen gesetzt, wie mit WIND_UPDATE und WIND_CALC.

Die Benutzung von Resource-Code-Files wurde durch verschiedene Befehle ebenfalls erleichtert: RSRC_LOAD, RSRC_FREE, RSRC_GADDR, RSRC_SADDR und RSRC_OBFIX lassen sich direkt als Befehle ansprechen. Abschließend soll hier noch die SHELL-Bibliothek erwähnt werden, auf die wir jedoch nicht näher eingehen möchten.

Zusammenfassung

GFA-BASIC V3.0 ist im Moment wohl der wohl leistungsfähigste BASIC-Interpreter auf dem ATARI ST und mit seiner Befehlsvielfalt von fast 400 Befehlen kaum zu überbieten (Es bleibt allerdings abzuwarten, was sich als Antwort im Hause OMIKRON tut). Für jede Anwendung lassen sich überaus nützliche Befehle einsetzen. Die Geschwindigkeit gegenüber Version V2.0 konnte im Interpreterbereich ebenfalls gesteigert werden. Der Editor überzeugt durch seine gut konzipierten Eigenschaften und durch die Syntaxüberprüfung jedes einzelnen Befehles direkt bei der Eingabe. Die Tastenbelegungen sind ebenfalls gut durchdacht und sparen dem Programmierer viel Zeit. Durch die Fehlersuchmöglichkeiten bleibt auch hier kaum ein Wunsch offen. Das Handbuch ist hervorragend aufgebaut und erläutert jeden Befehl äußerst ausführlich und genau, es ist stets ein Beispielprogramm abgedruckt, das die Funktion genau aufzeigt und die Syntax demonstriert. Vor allem auch die LINE-A und die AES-Implementation beeindrucken durch ihre einfache Benutzbarkeit und Vielfalt. Oft sind gleiche Befehle mit verschiedenen Worten ansprechbar (NEXT / ENDFOR), was sicher auch vielen Umsteigern von anderen Sprachen eine große Hilfe ist. Durch die 13-stellige Rechengenauigkeit kommen auch mathematisch Interessierte auf ihre Kosten.

Als negativ wurde von uns sehr wenig empfunden. Dazu gehört, daß die VDI-Routinen noch nicht komplett implementiert sind, so wie es mit den AES-Routinen geschehen ist. Hier wäre eine Implementation wünschenswert, zumindest von den Funktionen, die noch nicht durch jegliche Befehle benutzbar sind. Der passende Compiler soll in etwa einem halben Jahr erscheinen, es bleibt jedoch noch abzuwarten, wann dieser fehlerfrei zu haben sein wird. Aber es kann sich bis zur endgültigen Version ja noch einiges verändern. Inwieweit noch mehr Neuerungen eingebaut werden, bleibt abzuwarten, aber schon jetzt ist wirklich fast alles Denkbare vorhanden.

rp

Datenzeilen
DATA
READ
RESTORE
Dateiverwaltung
Verzeichnisse
DFREE()
CHDRIVE
DIRS
CHDIR
DIR TO
FILES TO
MKDIR
RMDIR
Dateien
EXIST
OPEN
LOF()
LOC()
EOF()
CLOSE
TOUCH
NAME AS
KILL
BLOAD
BSAVE
BGET
BPUT
Sequentieller/Indexseq. Zugriff
INP #
OUT #
INPUTS #
INPUT #
LINE INPUT #
STORE #
RECALL #
PRINT #
PRINT # USING
SEEK
RELSEEK
Wahlfreier Zugriff
FIELD AS
AT
RECORD
GET #
PUT #
Peripherie
Byteweise Ein- und Ausgabe
INP()
INP?()

Liste geht weiter...

OUT()
OUT?()

Serielle/MIDI Schnittstelle

INPAUXS
INPMIDS

Maus und Joystick

MOUSEX
MOUSEY
MOUSEK
MOUSE
SETMOUSE
HIDEM
SHOWM
STICK x
STICK()
STRIG

Drucker

LPRINT
LPOS()
HARDCOPY

Tonerzeugung

SOUND
WAVE

Entscheidungsanweisungen

IF
THEN
ELSE
ELSEIF
ENDIF

Mehrfachverzweigungen

ON GOSUB
SELECT CASE TO DEFAULT OTHERWISE
ENDSELECT CONT

Schleifen

FOR TO STEP NEXT ENDFOR
REPEAT UNTIL ENDREPEAT
WHILE WEND ENDWHILE
DO LOOP ENDDO DO WHILE DO UNTIL
LOOP WHILE LOOP UNTIL
EXIT IF

Unterprogrammbildung

GOSUB
PROCEDURE RETURN ENDPROC ENDSUB
LOCAL
VAR
FUNCTION ENDFUNC RETURN x DEFFN

Ereignisbezogene Verzweigungen

ON BREAK ON BREAK CONT ON BREAK GOSUB
ON ERROR ON ERROR GOSUB RESUME
RESUME NEXT
ERROR
ERR
ERRS
FATAL

Interrupt-Programmierung

EVERY EVERY STOP EVERY CONT
AFTER AFTER STOP AFTER CONT

Sonstige

REM
GOTO
PAUSE
DELAY
END
EDIT
STOP
SYSTEM
QUIT

Fehlerbehandlung

TRON
TROFF
TRON PROCEDURE
TRACES
DUMP

Grafik-Definitionsbefehle

SETCOLOR
COLOR
DEFMOUSE
DEFMARK
DEFFILL
BOUNDARY
DEFLINE
DEFTEXT
GRAPHMODE

Allgemeine Grafikbefehle

PLOT
LINE
DRAW
DRAW TO
BOX PBOX RBOX PRBOX
CIRCLE PCIRCLE ELLIPSE PELLIPSE
POLYLINE OFFSET POLYMARK OFFSET
POLYFILL OFFSET
POINT()
FILL
CLS
TEXT
SPRITE

Bildschirmausschnitte

SGET
SPUT
GET
PUT

Ereignisverwaltung

ON MENU MENU()
ON MENU BUTTON
ON MENU KEY GOSUB
ON MENU IBOX GOSUB
ON MENU OBOX GOSUB
ON MNEU MESSAGE GOSUB

Pull-Down-Menüs

ON MENU GOSUB
MENU m$()
MENU OFF
MENU KILL
MENU

Fensterbefehle

OPENW
CLOSEW
CLEARW
TITLEW
INFOW
TOPW
FULLW
WINDTAB

Sonstige

ALERT
FILESELECT

Systemroutinen

L: W: GEMDOS BIOS XBIOS

Line-A Aufrufe

PSET
PTST
ALINE
HLINE
ARECT
APOLY TO
BITBLT
ACHAR
ATEXT
ACLIP

VDI-Aufrufe

CONTRL INTIN PTSIN PTSOUT INTOUT
PTSOUT
VDISYS
VDIBASE
WORK_OUT()

GEM-AES Aufrufe

GCONTRL ADDRIN ADDROUT GINTIN
GINTOUT GB
GEMSYS

Unterroutinenaufruf

C:
MONITOR
CALL
RCALL
EXEC

Applikationsbibliothek (application library)

APPL_INIT
APPL_READ
APPL_WRITE
APPL_FIND
APPL_TPLAY
APPL_TRECORD
APPL_EXIT

Ereignis-Bibliothek (event library)

EVNT_KEYBD
EVNT_BUTTON
EVNT_MOUSE
EVNT_MESAG
EVNT_TIMER
EVNT_MULTI
EVNT_DCLICK

Menü-Bibliothek (menu library)

MENU_BAR
MENU_ICHECK
MENU_IENABLE
MENU_TNORMAL
MENU_TEXT
MENU_REGISTER

Objekt-Bibliothek (object library)

OBJC_ADD
OBJC_DELETE
OBJC_DRAW
OBJC_FIND
OBJC_OFFSET
OBJC_ORDER
OBJC_EDIT
OBJC_CHANGE

Formular-Bibliotheken (form library)

FORM_DO
FORM_DIAL
FORM_ALERT
FORM_ERRROR
FORM_CENTER
FORM_KEYBD
FORM_BUTTON

Grafik-Bibliothek (graphics library)

GRAF_RUBBERBOX
GRAF_DRAGBOX
GRAF_MOVEBOX
GRAF_GROWBOX
GRAF_SHRINKBOX
GRAF_WATCHBOX
GRAF_SLIDEBOX
GRAF_HANDLE
GRAF_MOUSE
GRAF_MKSTATE

Scrap-Bibliothek (scrap library)

SCRP_READ
SCRP_WRITE

Dateiauswahl-Bibliothek (file selector library)

FSEL_INPUT

Fenster-Bibliothek (window library)

WIND_CREATE
WIND_OPEN
WIND_CLOSE
WIND_DELETE
WIND_GET
WIND_SET
WIND_FIND
WIND_UPDATE
WIND_CALC

Resource-Bibliothek (resource library)

RSRC_LOAD
RSRC_FREE
RSRC_GADDR
RSRC_SADDR
RSRC_OBFIX

Shell-Bibliothek (shell library)

SHEL_READ
SHEL_WRITE
SHEL_GET
SHEL_PUT
SHEL_FIND
SHEL_ENVRN

ENDE

VOLLES



G RAMDISK II +

In der neuesten Version mit allen Leistungsmerkmalen, die man sich für eine Ramdisk überhaupt wünschen kann: Resetfest bis 4 MB, dabei abschaltbar; integrierter Druckerspooler, Bootcopy; Parametersave usw. Die G RAMDISK II ist im harten Einsatz bereits tausendfach bewährt.

DM 49,—



INTERPRINT II

Die universelle Anpassung für jeden Drucker hat einen integrierten Treiber, der auch alle Umlaute und Sonderzeichen nach Ihren Wünschen konvertiert. Dabei können mit Hilfe des Editors maßstabsgerechte Hardcopies auf allen Matrixdruckern und HP-Lasern erstellt werden. Auch ein Druckerspooler und unsere G RAMDISK II + sind integriert. Die Parameter sind speicherbar, und der komfortable Editor ermöglicht schnellste Anpassung.

DM 99,—

ohne Ramdisk **DM 49,—**



DISK HELP

Repariert Disketten und erlaubt in 99% der Fälle eine Wiederbelebung des gesamten Datenbestandes. Das Programm gehört neben jeden ST. Lassen Sie Ihre wichtigsten Daten nicht einfach von Laufwerken zerstören. DISK HELP ist einfach zu bedienen, korrigiert Lesefehler und rekonstruiert Files. Physikalische Fehler (Risse, Kaffee) können nicht behoben werden.

DM 79,—



Harddisk Help & extension

Wir wollen keinesfalls die Ängste um Ihre täglichen Daten schüren, aber sind Ihre Daten wirklich sicher? Wie schnell führt ein Headcrash zum Ende! Sie sollten auf ein bewährtes Backupsystem nicht länger verzichten. Auch sehr große Files können einzeln gesichert werden. Partition Backup, Tree Funktion, selected Backup und Diskoptimizer erhöhen die Effizienz dieses Programms. Auch bei der Geschwindigkeit wurden keine Kompromisse geschlossen.

DM 129,—



G COPY

Für Vielkopierer, Sicherheitskopierer, Raubkopierer und Umkopierer. Das universelle Kopierprogramm, das einfach mehr kann: G COPY kopiert alle ST-Programme und jeden Kopierschutz. G COPY formatiert alle Diskettenformate (80-87 Tracs, 9-11 Sektoren, 913 KB Disketten, Schnelladedisketten) in Hochgeschwindigkeit. Dabei erreicht G COPY eine unglaubliche Kopier- und Formatiergeschwindigkeit.
Jetzt neu: Mit Autocrack, Copy Protection, Detector und Copy Protection Emulator.

DM 99,—



FAST SPEEDER

Für alle, die ihrem Massenspeicher keine Pause gönnen wollen und lange Wartezeiten satt haben. Bringen Sie Ihre Festplatte und Diskettenstation auf Trab! Im Extremfall erreichen Sie dabei Ramdiskgeschwindigkeit! Festplatten werden bis zu 200% beschleunigt, Diskettenlaufwerke bis zu 900%. FAST SPEEDER ist einfach zu handhaben. Der optimierte Spezialalgorithmus mit Cache Prinzip arbeitet für Sie unmerklich im Hintergrund.

DM 129,—

G DISKMON II

Zum Kontrollieren, Ausprobieren, Umkopieren, Raubkopieren, Manipulieren, Interessieren und Reparieren. Ein nützliches und wichtiges Tool, damit die Diskette kein Geheimnis bleibt. Unbeschränktes Lesen, Schreiben, Editieren, Kopieren, Suchen, Ersetzen, Formatieren usw.; Schnelldruck; Bootsektorscannung in Klartextdarstellung; interaktive Hex, Dez oder ASCII Bedienung.

DM 99,—

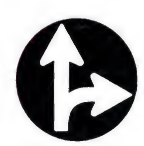


G DATEI

Maskengenerator, einfache, effektive Benutzerführung, Help Funktionen, Rechenfelder, Druckeranpassung, Listen- und Druckmaskengeneratoren sind für G DATEI selbstverständlich. Darüberhinaus sind in G DATEI ein Taschenrechner, eine kleine Textverarbeitung für Serienbriefe und ein Expertensystem integriert. Alles Logisch aufgebaut und ohne große Kenntnisse einfach zu bedienen. Damit Sie Ihre Daten nicht nur erfassen, sondern auch bearbeiten und auswerten können, wie Sie es wollen. Nicht umsonst ist G DATEI von der Gesellschaft der unabhängigen EDV-Berater empfohlen!

DM 199,—

Handbuch (vorab - wird angerechnet) **DM 20,—**



CHESS

Das stärkste Schachprogramm für ST Computer (bei 66.000 Eröffnungszügen) bietet nicht nur Spielstärke, sondern auch Komfort. Nachladbare und erweiterbare Eröffnungsbibliothek, beliebige Bedenkzeit (bei Ausnutzen Ihrer Bedenkzeit), Zugvorschlag, Seitenwechsel, Hängespiel, Problemschach, Patreindruck und Speicherung, Blitzen usw. sprechen klar für CHESS.

DM 129,—

16 BIT: AS SOUND SAMPLER III

NEU

Schon der Sound Sampler II überzeugte Kunden wie Presse (Keyboards 12/87: »Verglichen mit 10-15fach teureren Samplern... muß man von einer guten bis sehr guten Qualität sprechen.« ST 1/88: »Der Alleskönner.«). Alle Leistungsmerkmale des Samplers II bietet der neue Sound Sampler III in 16 Bit CD Qualität! Aufnahmen, editieren, verknüpfen, mischen, transponieren, Echoeffekte, einbinden in Programme, Sounds über MIDI Keyboards spielen; alles in original CD Qualität! Jetzt in 3 Versionen.

AS SOUND SAMPLER II Standard:	**DM 198,—**
AS SOUND SAMPLER II maxi:	**DM 298,—**
AS SOUND SAMPLER III 16 BIT:	**DM 598,—**
Soundbibliothek 8 BIT:	**DM 149,—**
Soundbibliothek 16 BIT:	**DM 198,—**

PROGRAMM



G SCANNER

Problemloses Digitalisieren von s/w-Bildern mit Ihrem Drucker ermöglicht der G SCANNER. Die hochwertige Abtastdiode von HP garantiert die ausgezeichnete Qualität von 200 dpi. Sie wird einfach auf dem Druckknopf Ihres Druckers befestigt. Mitgeliefert wird Software, die DEGAS und Normalbilder erstellt und Bildeditieren ermöglicht. Der G SCANNER ist voll kompatibel zu STAD und SYMPATIC PAINT (Scannoption).

DM 298,—

P6 Halterung für Diode DM 35,—

Neu: GSCANNER für Graustufen verwandelt Farbbilder in Graustufen. Kompatibel zu GSCANNER und STAD DM 398,—



Roboterarm

Der semiprofessionelle 5achsige Roboterarm wird anschlußfertig für ST, AMIGA oder PC/AT geliefert. Für Schulungszwecke, Lehr- und Demonstrationszwecke oder einfach zum Spielen bestens geeignet. Technische Daten: Höhe ca. 54 cm, industriegelb, ca. 2,5 kg schwer, einfacher Anschluß an Drukkerport. Auch über Joysticks zu steuern.

Anschlußfertig DM 269,—
Spezialnetzteil DM 99,—
Batteriesatz DM 10,—



TV Modulatoren

Schließen Sie Ihren ST an den Farbfernseher an, und sparen Sie sich einen teuren RGB Monitor. Unsere Modulatoren bringen ein scharfes Bild mit allen Mischtönen und einen Tonausgang extern und einen über den Fernseher. Das heißt: Bestes Bild und bester Tor für alle Fernseher.

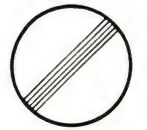
Typ A mit Monitorumschalter (ATARI-Fernseher) DM 249,—
Typ B ohne Monitorumschalter DM 239,—
Typ C FBAS Wandler für Geräte mit AV oder FBAS Eingang DM 169,—



RAM Erweiterungen

Unsere Ram Erweiterungen sind völlig neu. Sie werden ohne Lötarbeiten im Gerät gesteckt. Keine Leiterbahntrennungen – keine Treibersoftware. Die Platinen werden unbestückt geliefert und können mit 256 KBit Chips oder mit 1 MBit Chips bestückt werden. So erreicht man – je nach Wunsch – eine Erweiterung um 512 KB, 2 MB oder 4 MB. Dazu benötigen Sie nur eine Platine. Das bedeutet: zukunftssicher und problemlos.

für 520, 520+, 260, STM DM 249,—
für 1040 STF DM 249,—



LOGIC ANALYSER

Endlich ein Logik Analyser, den sich jedermann leisten kann. Dabei können wir auf die Leistungsmerkmale von 5-10fach teureren Geräten hinweisen: 16 Kanäle bis 600 KHz, komfortable GEM Software, verschiedene Triggerungsmöglichkeiten usw. Das Gerät wird auf den Romport gesteckt und anschlußfertig mit Software geliefert.

DM 448,—

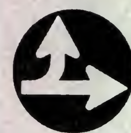
NEUHEITEN



RELAS

Aktienanalyse für den privaten und professionellen Aktienanleger auf der Basis der **Relativen Stärke und Volatilität** von Aktienkursen — **Charts** wahlweise über Bildschirm und Drucker — **Mischen von Charts** einer Aktie und Branche zum Vergleich — **Kurseingabe nur einmal wöchentlich (!)** Als Basis dienen die letzten 27 Wochenschlußkurse — Umfangreiche Bereinigungsfunktion bei Kapitalveränderungen von Aktien — **Listen** der Relativen Stärke mit Mittelwert von Kursen und Analyse der Veränderung der Rangfolge der Relativen Stärke über 3 Wochen (wahlweise Bildschirm oder Drucker) — Zu jeder Aktie können **Höchst- und Tiefstkurse** sowie eigene **Notizen** eingegeben werden — **Depotverwaltung** für den privaten und semiprofessionellen Aktienanleger über ein **bedienungsfreundliches Karteikartensystem** auf dem Bildschirm mit **Gewinn- und Verlustrechnung** Brutto — **Depotübersicht** wahlweise auf Bildschirm oder Drucker aufgeschlüsselt nach Einzelgattung oder als Gesamtübersicht — Umfangreiche **Lösch- und Änderungsfunktion** für die Daten

Handbuch vorab: DM 40,— DM 398,—



SYMPATIC PAINT

Der ATARI ST als Grafik Workstation der Superlative! Alle Funktionen, die für ein effektives und komfortables Zeichnen, Malen und Konstruieren erforderlich sind, wurden integriert. Der Benutzer hat nicht nur die Möglichkeit der Erstellung einer individuellen Grafiktoolbox, sondern auch eine Animationsselektion, mit der Bilder laufen lernen. Bilder können dabei mit dem G SCANNER und EASYTIZER eingelesen werden. Selbstverständlich kann die laufende Animation mit Sound unterlegt werden. Dabei können nicht nur alle ATARI Sound Register gezogen werden, sondern auch digitaler Sound vom AS SOUND SAMPLER II eingebaut werden. Trickfilme mit digitalem Sound!

Handbuch vorab: DM 30,— DM 298,—



GEM Retrace Recorder

Der Knüller! Wie von Geisterhand bewegt sich die Maus, laufen Kommandos ab, werden Aktionen gestartet. Der GEM Retrace Recorder ermöglicht es, jegliche Aktion (Mausbewegung, Menüs, Klicks, Tastatur usw.) aufzuzeichnen, und auf Tastendruck jederzeit original zu wiederholen. Das heißt: GEM Makros erstellen für Aktionen, die ständig wiederholt werden (auch in jedem Programm); super Auto Boots erstellen; selbstlaufende Demos von jedem Programm ohne Programmeingriffe und Kenntnisse u.v.m. Die Anwendungsmöglichkeiten sind fast unbegrenzt!

DM 99,—

Hardware Uhr

Die Uhr wird ohne Lötarbeiten im Rechner eingebaut, so daß der ROM Port frei bleibt. Sie haben ständig die aktuelle Zeit und das aktuelle Datum! Die Lithium Batterie garantiert hohe Ganggenauigkeit und 10 Jahre Laufzeit bei voller Schaltjahrerkennung. Der sensationelle Preis für die anschlußfertige Uhr beträgt nur

DM 79,—

G DATA

Siemensstraße 16 4630 Bochum 1
Verkaufsbüro:
Hattinger Str. 312, 4630 Bochum 1
Telefonische Bestellungen:
023 25 / 6 08 97

Schweiz: Senn Computer AG
Langstr. 31
CH-8021 Zürich

Österreich: Computershop Rittner
Hauptstr. 34
A-7000 Eisenstadt

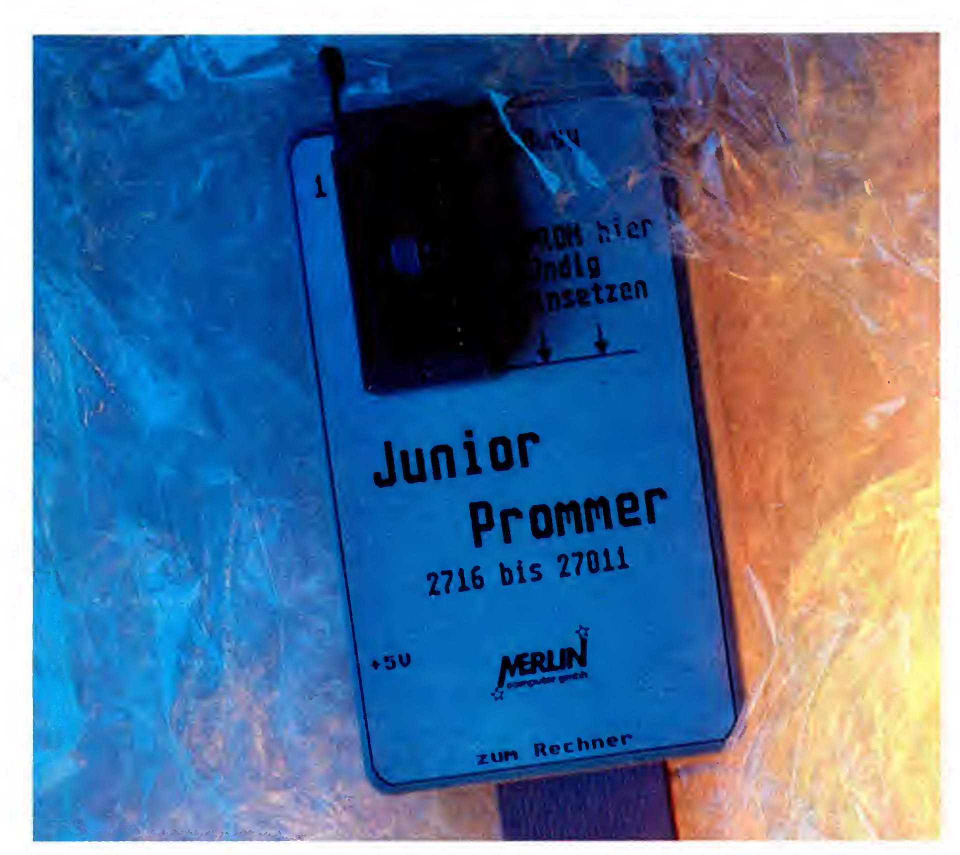
BENELUX:
G DATA BeNeLux
Postbus 70,
NL-2000 AB
Haarlem
Tel. 023 /
32 13 31

COUPON
Hiermit bestelle ich
☐ Informationsmaterial (DM 2,— beiliegend)
☐ 2 Demodisketten und Infomaterial (DM 10,— beiliegend)
☐ per Nachnahme (ca. DM 6,50 Versandkosten)
☐ Verrechnungsscheck liegt bei (+ DM 5,— Versandkosten)
Absender:

JUNIOR PROMMER

Die Software

Zum zweiten und letzten Mal geht es um den Juniorprommer. Diesmal wird jedoch fast ausschließlich die Software besprochen. Über ein vielseitiges Assemblerprogramm, welches Sie in beliebige, eigene Programme einbinden können, sind alle gängigen Operationen mit den Speicherbausteinen, wie z.B. Lesen, Programmieren, Vergleichen etc., möglich. Außerdem kann dieses Assemblerprogramm durch eine umfangreiche Konfigurationstabelle an die ausgefallendsten Speicherbausteine (auch Banking) angepaßt werden.



In der Februar-Ausgabe der ST-Computer wurde die Hardware und deren Programmierung beschrieben. Jetzt soll hauptsächlich ein Assemblerprogramm vorgestellt werden, welches aus Hochsprachen heraus benutzt werden kann, um alle Funktionen des Juniorprommers zu bedienen. Dieses Assemblerlmodul hat den Namen Juniorsoft und wird hier in Verbindung mit einem GFA-BASIC-Programm, genannt BEISPIEL.BAS vorgestellt. Die gegebenen Beispiele sind generell auch auf andere Hochsprachen übertragbar.

Betrieb der Hardware an ST-Rechnern

Da die Hardware flexibel ausgelegt wurde, soll der Anschluß speziell an ST-Rechner noch einmal erläutert werden. Zuerst benötigen Sie ein Verbindungskabel zur 5V-Spannungsversorgung des Juniorprommers. Dabei können Sie entweder ein externes, stabilisiertes Netzteil verwenden, oder Sie greifen mit lediglich einer Leitung die Spannungsversorgung ihres Rechners am Joystick-Port ab. Dazu benötigen Sie ein Verbindungskabel vom Joystickport Pin 7 zum Mittelstift der Stromversorgungsbuchse des Juniorprommers. Selbstverständlich entfällt die Steckbrücke zur Versorgung über die Schnittstelle. Jetzt ist der Juniorprommer, sofern fehlerfrei aufgebaut, betriebsbereit. Um die Funktionsfähigkeit zu prüfen, ist das Programm 'JUNIOR.BAS' abgedruckt, mit dem die Funktionen der Hardware überprüft werden sollen.

Juniorsoft

In Listing 1 ist der Quellcode Junior.q in seiner ursprünglichen Version für Profimat abgedruckt. Adaptieren auf ein anderes Entwicklungssystem ist, falls nötig, nicht mit Schwierigkeiten verbunden, da weder Bibliotheken noch spezielle Funktionen des Entwicklungssystems benutzt werden. Junior.q ist völlig lageunabhängig und sollte daher auch ohne TOS-Programmkopf assembliert werden. Junior.q beinhaltet folgende Funktionen:

1. Initialisieren des Juniorprommers für den weiteren Betrieb.
2. Bedienen der Schieberegister SRA bis SRD.

geht weiter...

3. Bedienen des Adreßzählers.
4. Kontrollieren der Datenrichtung.

Die weiteren Funktionen basieren auf den bisher genannten Funktionen.

5. Lesen des Speicherbausteins in den Speicher des Rechners.
6. Testen des Speicherbausteins, ob dieser leer ist. (Nur für EPROMs sinvoll.)
7. Vergleichen des Speicherbausteins mit dem Speicher des Rechners.
8. Prüfen, ob die Daten im Speicher des Rechners problemlos in den Speicherbaustein programmiert werden können. (Nur für EPROM's sinvoll.)
9. Programmieren eines Speichers.
10. Prüfen, ob ein Speicherbaustein des gewünschten Typs eingesetzt ist.
11. Speichertyp einstellen.
12. Initialisieren des Juniorprommers.

Das hier ausschließlich von 'Speicherbausteinen' und nicht von EPROMs geschrieben wird, hat folgenden Grund:
Über eine flexible Konfigurationstabelle kann nahezu jeder Speicherbaustein mit 28 oder 24 Anschlüssen über den Juniorprommer bedient werden. So ist es z.B. auch möglich, batteriegepufferte SRAMs zu programmieren, exotische ROMs zu lesen oder EEPROMs zu programmieren. Listing 2 zeigt die Typentabelle, welche bereits die EPROMs von 2716 bis 27011 enthält. Wie Sie neue Speichertypen definieren, wird noch erklärt. Es liegt in ihrem Ermessen diese Typentabelle beliebig zu erweitern.

Handhabung der Konfigurationstabelle in der Praxis

Falls Sie einen neuen Typ implementieren wollen, sollten Sie zuerst prüfen, ob dieser überhaupt geeignet ist. Die Steuerleitungen und restlichen Adreßleitungen sollten von SRA oder SRB abgedeckt werden, und die Adreß-, Daten-, Versorgungs- und Programmierspannungsleitungen, welche bereits fest verdrahtet sind, sollten an den festgelegten Pins liegen. Zu beachten ist, daß Speicher mit bis zu 17 Adreßleitungen möglich sind, wie z.B. die Rombausteine, die neuerdings in ST2/ST4 Modellen Verwendung finden. Nun füllen Sie am besten die Konfigurationstabelle mit Hilfe des Datenblat-

Funktionsübersicht

Ein Funktionsaufruf der Juniorsoft-Routinen aus GFA-BASIC heraus hat folgendes Grundgerüst:

Wert%=C:Junior%(Funktionsnummer,(L:)Parameter,(L:)Parameter,...)

Welche und wieviel Parameter Sie übergeben, hängt von den einzelnen Funktionen ab, die jetzt aufgezählt werden:

Funktion SRX,Nummer 0:

Aufruf: Void C:Junior%(0,Daten,Sr)

Parameter: Daten: Das Datenwort für das mit Sr ausgewählte Schieberegister.

Sr: Die Schieberegisternummer. 1 bis 3 entspricht SRA bis SRC.

Zweck: Transportiert den Parameter Daten in das mit SR spezifizierte Ausgangsschieberegister.

Funktion SRD,Nummer 1:

Aufruf: Byte%=C:Junior(1)

Parameter: Byte: Die Zustände der Datenleitungen D0-D7

Zweck: Gibt als Ergebnis die Zustände der 8 Datenleitungen des Speichers zurück.

Funktion AZ,Nummer 2:

Aufruf:Void C:Junior(2,Modus)

Parameter: Modus: Falls Modus = 0, wird der Adreßzähler gelöscht, sonst wird er inkrementiert.

Zweck: Steuerung des Adreßzählers.

Funktion SETZEA,Nummer 3:

Aufruf:Void C: Junior(3,Ea)

Parameter: *Ea*: Bestimmt die Datenrichtung von oder zum Speicherbaustein. Falls Ea = 0, wird der Speicherbaustein gelesen, bei Ea = 1 ist SRD auf die Datenleitungen geschaltet.

Zweck: Legt die Datenrichtung von oder zum Speicherbaustein fest.

Ab Funktionsnummer 4 wird zu deren Benutzung die Typentabelle benötigt. Wie diese Tabelle aufgebaut ist und erstellt werden kann, wird später beschrieben.

Funktion Lese,Nummer 4:

Aufruf: Ergebnis%=C:Junior(4,L:Ss,L:Es,L:Ee)

Parameter: *Ss*: Bestimmt die Speicherstartadresse im Rechner, ab der die Daten aus dem Speicherbaustein abgelegt werden.

Es: Die Startadresse im Speicherbaustein.

Ee: Die Endadresse im Speicherbaustein.

Ergebnis%: Enthält die letzte gelesene Adresse (=Ee).

geht weiter...

tes des Speichertyps aus. Einträge in der Konfigurationstabelle, die von Juniorsoft nicht explizit genutzt werden, können Sie von Ihrem eigenen Programm aus belegen. Zuletzt sollten Sie nicht vergessen, die LED auch in den verschiedenen Kontrollzuständen des EPROMs zu berücksichtigen. Die Konfigurationstabelle hat, wie schon gesagt, 82 Bytes pro Eintrag (Typ). Sie sollten immer darauf achten, daß genug freier Speicher für die Konfigurationstabelle vorhanden ist. Wohin Sie die Konfigurationstabelle im Arbeitsspeicher ablegen, muß immer den Gegebenheiten angepaßt werden. In GFA-BASIC bietet sich z.B. die Möglichkeit an, diese in einer Zeichenkette abzulegen, die vorher auf die geforderte Länge mit Leerzeichen aufgefüllt worden ist.

Die Juniorsoftfunktionen in der Praxis

Eine Beispielanwendung der Juniorsoft-Funktionen ist bereits in Listing 3 gezeigt.
Wie schon gesagt, dienen die Funktionen 0-3 zur Ansteuerung des Juniorprommers auf niedrigster Ebene. Die Funktionen 4-11 bauen auf diesen auf und ermöglichen die Ansteuerung des Juniorprommers wie einen Massenspeicher, benötigen zum Betrieb jedoch die Typentabelle. Es muß immer vor dem Aufruf dieser Funktionen ein gültiger Typ eingestellt werden, da es sonst zu Abstürzen kommen kann.
Die meisten Funktionen wie z.B. Lesen, Programmieren, benötigen einen Datenspeicher im Rechner. Auch dieser kann z.B. in einer Zeichenkette gehalten werden, jedoch nur bis 32768 Bytes.
Alle Adreßparameter der Funktionen werden nicht auf Gültigkeit geprüft, denn das liegt in dem Aufgabenbereich des Ansteuerprogramms. Außerdem sollten Adreßvariablen ausschließlich vom Typ Integer sein.
In Listing 4 ist ein Programm abgedruckt, das anhand des Kopierens eines EPROMs vom Typ 27128 fast alle Funktionen benutzt und erläutert.

Und zum Schluß

Falls Sie wenig Interesse daran haben, Ihr 'eigenes' EPROM zu programmieren, besteht auch noch die Möglichkeit, daß Sie eine fertige Software, ähnlich der des Easyprommers, über Merlin Computer GmbH beziehen. Auch die Hardware ist entweder nur als Platine oder als Fertiggerät erhältlich.

RH

Zweck: Mit dieser Funktion wird ein beliebiger Bereich des Speicherbausteins in den Speicher des Rechners gelesen.

Funktion Ltest, Nummer 5:

Aufruf:Ergebnis%=C:Junior%(5,L:Es,Ee)

Parameter: *Es*: Wie bei Funktion 4.
Ee: Wie bei Funktion 4.

Ergebnis%: Falls der Speicherbaustein im spezifizierten Bereich gelöscht ist, ist in Ergebnis% die letzte Adresse (=Ee). Ist der Speicherbaustein im spezifizierten Bereich nicht gelöscht, wird die fehlerhafte Adresse des Speicherbausteins mit gesetztem, höchstwertigen Bit zurückgegeben.

Zweck: Überprüfen des spezifizierten Bereichs im Speicherbaustein auf alle Datenbits = 1.

Funktion Vergl, Nummer 6:

Aufruf:Ergebnis%=C:Junior%(6,L:Ss,L:Es,L:Ee)

Parameter:Ss,Es,Ee sind definiert wie bei Funktion 4:

Ergebnis ist definiert wie bei Funktion 5, mit der Ausnahme, daß hier auf Gleichheit der Daten geprüft wird.

Zweck: Vergleicht den spezifizierten Speicherbereich des Speicherbausteins mit dem des Rechners.

Funktion Vtest, Nummer 7:

Aufruf:Ergebnis%=C:Junior%(7,L:Ss,L:Es,L:Ee)

Parameter: Alle Parameter definiert wie bei Nummer 6, mit der Ausnahme, daß hier geprüft wird, ob ein EPROM, das bereits programmiert ist, erneut mit den Daten ab Ss programmiert werden kann.

Zweck: Prüfen, ob ein erneutes Programmieren des Speichers möglich ist. Das ist möglich, da einzelne Bits eines EPROMs elektrisch auf Null programmiert werden können, jedoch nicht elektrisch auf 1. So wird hiermit lediglich geprüft, daß die aktuellen Daten kein Bit auf 1 setzen, welches im EPROM schon auf Null gesetzt ist.

Funktion Ipro,Nummer 8:

Aufruf:Ergebnis%=C:Junior%(8,L:Ss,L:Es,L:Ee,Flagge)

Parameter: Ss,Es,Es und Ergebnis% sind definiert wie bei Nummer 6, mit dem Unterschied, daß Ergebnis die erste fehlerhafte Adresse enthält. Ist Flagge = 0, werden alle $FF-Bytes bei der Programmierung übergangen, was nur bei EPROMs sinnvoll ist. Ist Flagge jedoch 1, werden alle Bytes programmiert.

Zweck: Programmiert einen Speicherbaustein in dem spezifizierten Bereich mit dem spezifizierten Speicherbereich des Rechners. Ausgehend von dem standardisierten 'intelligenten' Programmieralgorithmus, wie er in verschiedenen Datenblättern vorgeschlagen wird, können alle anderen Algorithmen durch Parametervariation in der Konfigurationstabelle von diesem abgeleitet werden.

geht weiter...

Funktion Tstest, Nummer 9:

Aufruf: Ergebnis%=C:Junior%(9)

Parameter: Ergebnis% ist -1, falls der Speicherbaustein keine gültige Rückmeldung gibt, also nicht oder nicht richtig eingesetzt ist. Ist Ergebnis% = 0, dann ist höchstwahrscheinlich ein funktionsfähiger Speicherbaustein eingesetzt.

Zweck: Prüfen, ob ein Speicher eingesetzt ist.

Funktion Waltp, Nummer 10:

Aufruf: Void C:Junior%(10,L:Typ%)

Parameter: Typ% ist die Adresse, an der die dem Typ entsprechende, 82 Bytes lange, Konfigurationstabelle stehen muß.

Zweck: Einstellen des Speichertyps.

Funktion Init, Nummer 11:

Aufruf: Void C:Junior%(11)

Keine Parameter.

Zweck: Initialisiert den Juniorprommer anfangs für den weiteren Betrieb.

Die Typenkonfigurationstabelle

Wie Sie jetzt wissen, können Sie fast jeden 28- oder 24-poligen Speicherbaustein mit Hilfe dieser Tabelle in den Juniorprommer einbinden. In dieser Tabelle sind alle möglichen Daten von der Typenbezeichnung bis zur Programmierspannung enthalten. Listing 2 zeigt diese Typentabelle bereits erstellt für die EPROMs 2716 bis 27011. Ein kompletter Eintrag, welcher jeweils einen Speichertyp repräsentiert, ist genau 82 Bytes lang, muß an einer geraden Adresse beginnen und ist so aufgebaut:

1. Eintrag, Zkette

Offset: 0

Datentyp: Zeichenkette mit konstant 30 Zeichen.

Die Zeichenkette dient lediglich der Benutzerorientierung und beschreibt den Speichertyp in Kurzform.

2. Eintrag, Kapazi

Offset: 30

Datentyp: Langwort.

Kapazi muß die Kapazität des Speichertyps in Bytes enthalten.

3. Eintrag, Seite

Offset: 34

Datentyp: Langwort.

Seite muß die Seitengröße eines gebankten Speichers in Byte enthalten. Falls der zu spezifizierende Speicher nicht gebankt ist, muß Seite = 0 sein.

4. Eintrag, Alg

Offset: 38

Datentyp: Wort.

Dient lediglich der Benutzerorientierung und kann einen Wert enthalten, der Aufschluß über die möglichen Programmieralgorithmen gibt.

5. Eintrag, I_Zeit

Offset: 40

Datentyp: Wort

Bestimmt die Programmierdauer des Initialprogrammierimpulses in 0,1 ms Schritten.

6. Eintrag, M_Zeit

Offset 42

Datentyp: Wort.

Bestimmt die Zeitdauer, welche mit der Anzahl der Initialimpulse multipliziert die Nachprogrammierdauer ergibt, in 0.1ms Schritten. Ist M_Zeit = 0, dann wird das Nachprogrammieren übergangen.

7. Eintrag, Trials

Offset 44

Datentyp: Wort.

Legt die Anzahl der Initialprogrammierversuche fest, nach denen die Programmierung abgebrochen wird.

8. Eintrag, Standby

Offset: 46

Datentyp: 2 Bytes.

Enthält die Kontrollbytes für den spezifischen Betriebszustand Standby des Speichers. Dieser Betriebszustand ist wie alle in den Datenblättern der Hersteller beschrieben. Das erste Kontrollbyte korrespondiert direkt mit SRA, das zweite direkt mit SRB.

9. Eintrag, Read

geht weiter...

Offset 48

Datentyp: 2 Bytes.

Wie oben, jedoch wird der Betriebszustand Read beschrieben.

10. Eintrag, Odis

Offset 50

Datentyp 2 Bytes.

Bestimmt den Betriebszustand 'Output Disable'.

11. Eintrag, Program

Offset 52

Datentyp 2 Bytes.

Bestimmt den Betriebszustand 'Program'.

12. Eintrag, Pr_ver

Offset 54

Datentyp 2 Bytes.

Bestimmt den Betriebszustand 'Program Verify'.

13. Eintrag, Pr_inh

Offset 56

Datentyp 2 Bytes.

Bestimmt den Betriebszustand 'Program Inhibit'.

14. Eintrag, Page_wr

Offset 58

Datentyp 2 Bytes.

Bestimmt den Betriebszustand 'Page Write'.

15. Eintrag, Manufac

Offset 60

Datentyp Byte.

Hat hier keine Bedeutung, würde jedoch den Herstellercode des Speichers angeben.

16. Eintrag, Devcode

Offset: 62

Datentyp: Byte.

Wie oben, jedoch würde dieses Byte den Typencode angeben.

17. Eintrag, Tsmask

Offset 62

Datentyp: Byte.

Hat hier keine Bedeutung.

18. Eintrag, Frei3

Offset 63

Datentyp: Byte.

Unbenutzter Eintrag.

19. Eintrag, Frei1

Offset 64

Datentyp: Wort.

Unbenutzter Eintrag.

20. Eintrag, Frei2

Offset 66

Datentyp: Wort.

Unbenutzter Eintrag.

21. Eintrag, M_a10

Offset 68

Datentyp: 2 Byte.

Bestimmt den Pin, der bei dem zu beschreibenden Speicher mit A10 benannt ist. Das erste Byte korrespondiert hier wieder direkt mit SRA, das zweite Byte direkt mit SRB. Da diese Daten zu den Betriebszuständen oderverknüpft werden, müssen die Betriebszustände für alle spezifizierten Adreßpins Null sein.

22. Eintrag, M_a11

Offset 70

Datentyp: 2 Byte.

Wie oben, jedoch für A11.

Die Tabelle wird chronologisch bis zum 27. Eintrag, der A16 beschreibt, fortgeführt.
In Listing 2 finden Sie Beispiele mit den Eprom-Typen 2716 bis 27011.

Die Listings folgen auf den nächsten Seiten.

Listing 1:
Der Assemblerquellcode 'JUNIOR.Q' der Juniorsoft-Funktionen in seiner Version für Profimat.

```
 1:  ;              JUNIOR.Q - Assemblerquellcode
 2:  ;  zur Ansteuerung des Juniorprommers Lageunabhängiges
 3:  ;      Assemblermodul zur Einbindung in Hochsprachen
 4:  ;    Parameterübergabe auf dem Stack nach C-Konventionen
 5:  ;                    R.Hofmann 1/88
 6:
 7:  mfport  = $fffffa01
 8:
 9:  psgrd   = $ffff8800
10:  psgwr   = $ffff8802
11:  max_f   = 11
12:
13:  strobe  = %00100000
14:
15:  azb     = %00010000
16:  lazb    = %01000000
17:
18:  bpe     = 82
19:  ;            Indizies in die Typentabelle
20:  ; 82 Bytes/Eintrag
21:  zkette  = 0
22:  kapazi  = 30
23:  seite   = 34
24:  alg     = 38
25:  i_zeit  = 40
26:  m_zeit  = 42
27:  trials  = 44
28:  standby = 46
29:  read    = 48
30:  odis    = 50
31:  program = 52
32:  pr_ver  = 54
33:  pr_inh  = 56
34:  page_wr = 58
35:  manufac = 60
36:  devcode = 61
37:  tsmask  = 62
38:  frei3   = 63
39:  frei1   = 64
40:  frei2   = 66
41:
42:  m_a10   = 68
43:  m_a11   = 70
44:  m_a12   = 72
45:  m_a13   = 74
46:  m_a14   = 76
47:  m_a15   = 78
48:  m_a16   = 80
49:
50:          text
51:
52:  funktion movem.l d1-d7/a0-a6,-(sp) ; Hauptverteiler
53:          lea funktion(pc),a6     ; Relative Basisadresse
54:          lea 62(a7),a5            ; Parameterbasis
55:          move.l a5,parbas(a6)     ; "
56:          clr.l -(sp)
57:          move.w #$20,-(sp)        ; SUPER
58:          trap #1                  ; GEMDOS
59:          addq.l #6,sp             ; ab hier im
                                        Supervisormodus
60:          move.l d0,alt_ssp(a6)    ; SSP retten
61:
62:          lea stopikb(a6),a0       ; IKBD:Stop
63:          move.l a0,-(sp)
64:          clr.w -(sp)
65:          move.w #25,-(sp)
66:          trap #14
67:          addq.l #8,sp
68:          move sr,alt_sr(a6)       ; SR retten
69:          move.w #100,d0           ; 0.01 s warten
70:          bsr zeit
71:          ori #$0700,sr            ; IRQ's abschalten
72:
73:          bsr direkt               ; Funktion aufrufen
74:          move.l d0,-(sp)
75:          move alt_sr(a6),sr
76:          lea startikb(a6),a0      ; IKBD:Start
77:          move.l a0,-(sp)
78:          clr.w -(sp)
79:          move.w #25,-(sp)
80:          trap #14
81:          addq.l #8,sp
82:
83:          move.l alt_ssp(a6),-(sp)
84:          move.w #$20,-(sp)
85:          trap #1
86:          addq.l #6,sp              ; und im User-Modus
                                         zurückkehren
87:          move.l (sp)+,d0
88:          movem.l (sp)+,d1-d7/a0-a6
89:  ende    rts
```

```
 90:
 91:
 92:  direkt  move.w -2(a5),d0           ; Funktionsnummer
 93:          tst.w d0                   ; im legalen Bereich ?
 94:          bmi.s ende
 95:          cmp.w #max_f,d0
 96:          bhi.s ende
 97:          lsl.w #2,d0                ; zu Zeiger
 98:          move.l f_adr(pc,d0.w),a0   ; Funktionsadresse
 99:          add.l a6,a0                ; Offset
100:          jmp (a0)                   ; Funktion anspringen
101:
102:
103:  ;              Zeiger auf Funktionen
104:  f_adr   dc.l srx                   ; 0=SRX
105:          dc.l srd                   ; 1=SRD
106:          dc.l az                    ; 2=AZ
107:          dc.l setzea                ; 3=SETZEA
108:          dc.l lese                  ; 4=LESEN
109:          dc.l ltest                 ; 5=LTEST
110:          dc.l vergl                 ; 6=VERGL
111:          dc.l vtest                 ; 7=VTEST
112:          dc.l ipro                  ; 8=PROG
113:          dc.l tstest                ; 9=TRTEST
114:          dc.l waltp                 ; 10=TYP
115:          dc.l init                  ; 11=INIT
116:
117:  sra     movem.l d0/d2,-(sp)
118:          moveq #2,d1
119:          or.b adrsra(a6),d0
120:          move.w d0,-(sp)
121:          bsr srxd                   ; Byte herausschieben
122:          move.w (sp),d0             ; alter Wert
123:          move.b d0,d1
124:          and.b #%11100000,d0        ; Testen auf Änderung
125:          move.b letzt_a2(a6),d2
126:          and.b #%11100000,d2
127:          cmp.b d0,d2
128:          beq.s ta15                 ; Alles gleich

129:          scs d2
130:          move.w #1000,d0            ; 0.1s Verzögerung
131:          tst.b d2
132:          bne.s zver
133:          move.w #8000,d0            ; 0.8s Verzögerung
134:  zver    bsr zeit
135:  ta15    and.b #1,d1                ; A15 auf Änderung
                                           testen
136:          move.b letzt_a2(a6),d0
137:          and.b #1,d0
138:          cmp.b d0,d1
139:          beq.s uspr                 ; Alles gleich

140:          moveq #50,d0               ; 5 mS Verzögerung
141:          bsr zeit
142:  uspr    move.w (sp)+,d0            ; Alter Vergleichswert
143:          move.b d0,letzt_a2(a6)     ; als Byte
144:          movem.l (sp)+,d0/d2
145:          move.b d0,letzt_a(a6)
146:          rts
147:
148:  srb     moveq #4,d1               ; Taktbit
149:          move.b d0,letzt_b(a6)
150:          or.b adrsrb(a6),d0        ; Adressen
151:          eori.b #%01000000,d0      ; A13 invers
152:          bra.s srxd
153:
154:  src     moveq #8,d1                ; Datenschieberegister
155:          bra.s srxd
156:
157:  srx     move.w (a5),d0             ; Daten
158:          moveq #1,d1
159:  decoder lsl.w #1,d1
160:          subq.w #1,2(a5)           ; Takt
161:          bne.s decoder
162:
163:  ;       Bedient Schieberegister SRA bis SRC
164:  srxd    movem.l d2-d3/a0,-(sp)
165:          lea psgwr,a0
166:          move.b #15,-2(a0)         ; Portb
167:          move.w rwstat(a6),d3
168:          move.w d3,d2              ; 1.Bit
169:          roxl.b #1,d0
170:          roxr.b #1,d2
171:          move.b d2,(a0)
172:          or.b d1,d2
173:          move.b d2,(a0)
174:          move.w d3,d2              ; 2.Bit
175:          roxl.b #1,d0
176:          roxr.b #1,d2
177:          move.b d2,(a0)
178:          or.b d1,d2
179:          move.b d2,(a0)
180:          move.w d3,d2              ; 3.Bit
181:          roxl.b #1,d0
```

```
182:          roxr.b #1,d2
183:          move.b d2,(a0)
184:          or.b d1,d2
185:          move.b d2,(a0)
186:          move.w d3,d2                ; 4.Bit
187:          roxl.b #1,d0
188:          roxr.b #1,d2
189:          move.b d2,(a0)
190:          or.b d1,d2
191:          move.b d2,(a0)
192:          move.w d3,d2                ; 5.Bit
193:          roxl.b #1,d0
194:          roxr.b #1,d2
195:          move.b d2,(a0)
196:          or.b d1,d2
197:          move.b d2,(a0)
198:          move.w d3,d2                ; 6.Bit
199:          roxl.b #1,d0
200:          roxr.b #1,d2
201:          move.b d2,(a0)
202:          or.b d1,d2
203:          move.b d2,(a0)
204:          move.w d3,d2                ; 7.Bit
205:          roxl.b #1,d0
206:          roxr.b #1,d2
207:          move.b d2,(a0)
208:          or.b d1,d2
209:          move.b d2,(a0)
210:          move.w d3,d2                ; 8.Bit
211:          roxl.b #1,d0
212:          roxr.b #1,d2
213:          move.b d2,(a0)
214:          or.b d1,d2
215:          move.b d2,(a0)
216:
217:          move.b #14,-2(a0)           ; Port a
218:          move.b -2(a0),d2
219:          or.b #strobe,d2             ; Strobe = 1
220:          move.b d2,(a0)
221:          and.b #255-strobe,d2        ; Strobe = 0
222:          move.b d2,(a0)
223:          movem.l (sp)+,d2-d3/a0
224:          rts
225:
226:  srd:
227:
228:  ; Bedient srd
229:  srdd    movem.l d3-d4/a1-a2,-(sp)  ; Eingangsschiebe-
                                         Register lesen
230:          lea psgwr,a1
231:          move.b #14,-2(a1)           ; Psg Port A
232:          move.b (a1),d3              ; Strobeimpuls geben
233:          or.b #strobe,d3
234:          move.b d3,(a1)
235:          lea mfport-1,a2
236:          and.b #255-strobe,d3
237:          move.b d3,(a1)
238:          moveq #32,d3                ; Register vorbesetzen
239:          moveq #0,d4
240:          moveq #0,d0
241:          move.b #15,-2(a1)           ; Psg Port B
242:
243:          roxr.w (a2)
244:          move.b d3,(a1)
245:          addx.w d0,d0
246:          move.b d4,(a1)
247:
248:          roxr.w (a2)
249:          move.b d3,(a1)
250:          addx.w d0,d0
251:          move.b d4,(a1)
252:
253:          roxr.w (a2)
254:          move.b d3,(a1)
255:          addx.w d0,d0
256:          move.b d4,(a1)
257:
258:          roxr.w (a2)
259:          move.b d3,(a1)
260:          addx.w d0,d0
261:          move.b d4,(a1)
262:
263:          roxr.w (a2)
264:          move.b d3,(a1)
265:          addx.w d0,d0
266:          move.b d4,(a1)
267:
268:          roxr.w (a2)
269:          move.b d3,(a1)
270:          addx.w d0,d0
271:          move.b d4,(a1)
272:
273:          roxr.w (a2)
274:          move.b d3,(a1)
275:          addx.w d0,d0
```

```
276:          move.b d4,(a1)
277:
278:          roxr.w (a2)
279:          move.b d3,(a1)
280:          addx.w d0,d0
281:
282:          not.b d0                ; Datenwort invertieren
283:          movem.l (sp)+,d3-d4/a1-a2
284:          rts
285:
286:
287:  az      tst.w (a5)              ; Adreßzähler inkrement.
288:          beq lazd                ; /löschen
289:
290:  ;             Adreßzähler erhöhen
291:  azd     move.b #15,psgrd        ; Psg port B
292:          move.w rwstat(a6),d0
293:          lsr.w #1,d0
294:          ori.b #azb,d0
295:          move.b d0,psgwr
296:          and.b #255-azb,d0
297:          move.b d0,psgwr
298:          rts
299:  ;             Adreßzähler löschen
300:  lazd    move.b #15,psgrd        ; Psg Port B
301:          move.w rwstat(a6),d0
302:          lsr.w #1,d0
303:          or.b #lazb,d0
304:          move.b d0,psgwr
305:          and.b #255-lazb,d0
306:          move.b d0,psgwr
307:          rts
308:
309:  setzea  move.w (a5),d0          ;  Eingabe-/Ausgabestatus
310:          and.w #1,d0             ;  setzen
311:
312:  ;                 setzt E/A Status
313:  setzead move.b #15,psgrd
314:          move.b d0,psgwr
315:          lsl.w #1,d0
316:          move.w d0,rwstat(a6)
317:          rts
318:
319:  ;setzt Adresse in d0 auf den virtuellen Speicher
320:  setzadr movem.l d1-d4/a0/a1,-(sp)
321:          move.l d0,d2
322:          move.l typzeig(a6),a1
323:  wdh     cmp.l istadr(a6),d2
324:          beq.s stimmt
325:          bhi.s gr
326:          clr.l istadr(a6)
327:          bsr lazd
328:          bra nerh
329:  gr      addq.l #1,istadr(a6)        ; Adresse erhöhen
330:          bsr azd                     ; Adreßzähler auch
331:  nerh    move.l istadr(a6),d1
332:          lsr.l #8,d1
333:          lsr.l #2,d1
334:          cmp.l letzta10(a6),d1       ; gleiche Adresse ??
335:          beq.s wdh                   ; Ja, dann unnötig
336:          tst.l seite(a1)             ; gebankter Speicher
337:          beq.s kbea
338:          move.l istadr(a6),d3
339:          divu seite+2(a1),d3         ; Seite errechnen
340:          cmp.w aseite(a6),d3         ; gleiche wie zuvor?
341:          beq.s kbea
342:          move.w d3,aseite(a6)
343:          bsr seitesz
344:  kbea    move.l d1,letzta10(a6)      ; jetzt gleich
345:          move.l a1,a0
346:          add.w #m_a10,a0
347:          moveq #6,d0                 ; maximal 7 Adreßpin
348:          clr.w d3
349:          clr.w d4
350:  setzax  lsr.w #1,d1                 ; Adresse
351:          bcc.s axnull
352:          or.b (a0),d3                ; SRA
353:          or.b 1(a0),d4               ; SRB
354:  axnull  addq.l #2,a0
355:          dbra d0,setzax
356:          move.b d3,adrsra(a6)        ; Oder-Adressen
357:          move.b d4,adrsrb(a6)
358:          move.b letzt_a(a6),d0       ; Adresse
359:          bsr sra                     ; aktualisieren
360:          move.b letzt_b(a6),d0
361:          bsr srb
362:          bra wdh
363:  stimmt  movem.l (sp)+,d1-d4/a0/a1
364:          rts
365:
366:  seitesz: movem.l d0-d3/a0-a1,-(sp)
367:          move.l typzeig(a6),a1
368:          move.w aseite(a6),d0      ; Aktuelle Speicherbank
369:          bsr src                   ; auf den Datenbus
370:          moveq #1,d0
```

720 KB 3.5 Zoll NEC FD 1037a Laufwerk anschlußfertig 279,-

Superflaches Diskettenlaufwerk (25.4 mm hoch) elegantes, sehr kratzfestes Stahlblechgehäuse



Steckernetzteil
5V stabilisiert, eigens für NEC FD 1037 entwickelt
In unseren anschlußfertigen Laufwerken serienmäßig enthalten **34,90**

Floppygehäuse Stahlbl. Atari grau
Für die Laufwerkstypen NEC 1036a, NEC 1037a, Fd 55 FR., kompl. mit Zugentlastung und Füßchen **29,-**

5.25 Laufwerk 720 KB

- anschlußfertig m. 40/80 Tr. Umsch.
- beige Frontblende o. Aufpreis
- incl. Software IBM-ATARI
- TEAC FD 55 FR

349,-

Einbau 2. Floppybuchse + 27,90

Umschaltung Drive A/B + 20,-
nur in Verbindung mit 2. Floppybuchse erhältlich. Ermöglicht das wahlweise Booten von Laufwerk A oder Laufwerk B

NEC FD 1037a 197,-

Teac FD 55 FR 229,- modif. 239,-

Switchbox ab 29,90

Umschalter SW/Farbe



Switchbox mit Monitorkabel 50 cm lang. Audioausgang o. Abb.
Bausatz **29,90**
Fertiggerät f. Mega ST **39,90**
Switchbox direkt an den Rechner anstöpselbar (daher kein Kabelsalat mehr)
siehe Abb. **49,90**

Floppyswitchbox 89,-

Ermöglicht den Anschluß von 3 Laufwerken am ST
Das Umschalten erfolgt ohne Reset. Die Switchbox ist mit Treiber und Original Buchsen ausgestattet.

Druckerswitchbox **89,-**
1 Drucker an 2 Computer oder 2 Computer an einen Drucker

Tastaturgehäuse **124,-**
Lieferung komplett mit Spiralkabel Resettaster und Joystickbuchsen eingebaut

Blitter TOS o. Rom TOS **109,-**
mit oder ohne Fast Load

Monitorverlängerung NEU 29,90

Floppyverlängerung NEU 29,90

Fast ROM o. Blittertos -Fastrom o. Fast Load Ch. Hertz **24,90**

Speichererweiterung **84,90**
Leerkarte bestückt o. Rams

Uhrmodul (extern) 89,-

Uhrmodul (intern) **129,-**
integrierte Bootsoftware auf Roms

Stecker und Buchsen
Monitorstecker **6,40** Monitorbuchse **8,90**
Floppystecker **8.90** Floppybuchse **10.90**

Kabel
Floppykabel Atari — 3.5 Zoll (Shugart) **27,90**
Audiokabel Switchbox ST ca. 2.5 m **7,90**
Scartkabel, 2 m **39,90** Druckerkabel, 2 m **29,90**

Neuheiten ST Software

Automon **74,-**
Softwaremäßige Umschaltung zwischen SW- u. Farbmonitor. Umschaltung erfolgt ohne Reset.

AT Tastatur am ST **189,-/289,-**
Adapterbox zum Anschluß einer AT Tastatur am ST
Einbautyp/Fertiggerät

Signum II	Preis auf Anfrage
STAD	Preis auf Anfrage
Omikron. Basic	159,-
Omikron. Comp.	159,-
PC Ditto	197,-

Public-domain (alle ST Disketten)
incl. Diskette eins. format. DM 6,40
Liste gegen DM 2,- in Briefmarken

1 Paket — 5 Disk. doppels. format. nur 35,-

- **P.1 Die besten Spielprogr.** z. B. Solitaire, Metropol, Mac Pan .....
- **P.2 Die besten Utilitieprogr.** z. B. Fcopy, Diskchecker, Ramdisk .....
- **P.3 Die besten Mal- u. Soundp.** z. B. E-Plan, Soundcomputer, Andromeda .....
- **P.4 Die besten diversen Prg.** z. B. Etikettdat., Aktien, ST-Calc .....

Hard- u. Soft A. Herberg, Bahnhofstr. 289, 4620 Castrop-Rauxel, Tel. (0 23 05) 1 57 64 —Händleranfragen erwünscht—

SPC MODULA-2

für ATARI ST und MEGA ST

Warum Modula-2 ?

Das Hauptproblem der Softwareherstellung ist die **Wartung** der Produkte. Mit jeder Programmiersprache läßt sich eine große Menge Programm in kurzer Zeit einhacken. In dem Moment, wo Fehler offensichtlich sind oder Änderungswünsche auftauchen, wird der Vorteil von Modula-2 sichtbar. Teilaufgaben werden in **unabhängige Module** verpackt. Viele Programmierer können gleichzeitig an einem Programm arbeiten. Auch Module aus fremder Quelle oder früheren Projekten sind ohne Probleme einsetzbar.

Warum SPC Modula-2 ?

SPC MODULA-2 ist die direkte Umsetzung der **neuesten Version** des an der ETH Zürich entwickelten Compilers. Unser Ziel ist eine komplette Entwicklungsumgebung **'aus einem Guß'**. Ein syntaxunterstützender Editor und ein symbolischer Debugger sind dabei nur die Minimalforderung. Unsere systemunabhängige Windowschnittstelle **SSWiS** und ein mächtiges **MAKE Utility** sind weitere Schritte auf diesem Weg. Viele weitere werden folgen.

LEISTUNGSMERKMALE

- Ein Compiler mir einer Übersetzungsleistung von 5000 Zeilen pro Minute; **neuer Wirth'scher Standard**
- Ein Editor, der bis zu **8 Dateien** gleichzeitig editiert, und die MODULA-Syntax unterstützt.
- Ein **symbolischer Debugger**, der bei Programmfehlern automatisch aufgerufen wird.
- Das lästige Binden von Programmen entfällt; SPC MODULA-2 ist ein **Single Pass Compiler**
- Der **Editierzyklus** ist kürzer als eine Minute
- COROUTINEN werden unterstützt
- Versionen für RTOS und OS/9 in Vorbereitung
- Eine portable Windowschnittstelle: **SSWiS**
- **In Vorbereitung: Datenbank ADIPROG**

DM 348,-

ADVANCED APPLICATIONS
Viczena GmbH
Sperlingweg 19
D-7500 Karlsruhe 31
Tel.: (0721) 70 09 12

DEMODISKETTE für DM 10,00 anfordern

```
371:          bsr setzead              ; schreiben
372:          move.b letzt_a(a6),d2    ; SRA
373:          move.b letzt_b(a6),d3    ; SRB
374:          move.b d2,d0
375:          andi.b #%11100111,d0     ; evtl. Progammiersp.
376:          bsr sra                  ; ausschalten
377:          move.w #100,d0
378:          bsr zeit                 ; 10 mS Warten
379:          move.b odis+1(a1),d0
380:          bsr srb
381:          move.b page_wr+1(a1),d0
382:          bsr srb
383:          move.b odis+1(a1),d0
384:          bsr srb
385:          move.b d2,d0
386:          bsr sra
387:          move.b d3,d0
388:          bsr srb
389:          move.w #20,d0
390:          bsr zeit                 ; 2mS Warten
391:          moveq #0,d0
392:          bsr setzead
393:          movem.l (sp)+,d0-d3/a0-a1
394:          rts
395:
396: sraus    move.b d2,d0
397:          bsr sra
398:          move.b d3,d0
399:          bra srb
400:
401:  ;          Legt Kontrollzustände bei A0 an den
                 virtuellen Speicher
402: setzst:  move.l a0,-(sp)
403:          move.b (a0)+,d0
404:          cmp.b letzt_a(a6),d0
405:          beq.s ubf1
406:          bsr sra                  ; Schieberegister a
407: ubf1     move.b (a0)+,d0
408:          cmp.b letzt_b(a6),d0
409:          beq.s ubf2
410:          bsr srb
411: ubf2     move.l (sp)+,a0
412:          rts
413:
414:  ;                     Zeitverzögerung
415: zeit:    beq.s nullzt
416:          mulu #40,d0              ; in 0.1ms Schritten
417:          sub.l #38,d0
418: schleif  subq.l #1,d0             ; 12 Takte
419:          bne.s schleif            ; 8 Takte
420: nullzt   rts
421:
422:
423:  ; Aktiviert den Speicher und löscht die Adresse
424: einsch:  clr.l adrsra(a6)
425:          clr.l letzta10(a6)
426:          clr.w letzt_a(a6)
427:          clr.w letzt_a2(a6)
428:          clr.w letzt_b(a6)
429:          clr.w aseite(a6)
430:          moveq #0,d0          ; Ausgangsschieberegister aus
431:          bsr setzead
432:          move.l typzeig(a6),a0 ; Aktuelles EPROM
433:          add.w #odis,a0        ; Betriebszustand
434:          bsr setzst
435:          bsr seitesz          ; Falls nötig, Seite löschen
436:          clr.l d0              ; Adresse zurücksetzen
437:          bra setzadr
438:
439: ; Inaktiviert den Speicher und löscht die Adresse
440: aussch:  moveq #0,d0
441:          bsr setzead
442:          move.l typzeig(a6),a0
443:          add.w #standby,a0
444:          bsr setzst
445:          clr.l d0
446:          bra setzadr
447:
448:  ;       Liest den Speicher in den des Rechners
449:  ; Parameter : 0/Adresse.l,4/EPROMstart,8/EPROMende
450: lese:    move.l (a5),a2           ; Speicheradresse
451:          move.l 4(a5),d3          ; EPROMstart
452:          move.l 8(a5),d4          ; EPROMende
453:          bsr einsch               ; Speicher aktivieren
454:          move.l typzeig(a6),a0
455:          add.w #read,a0
456:          bsr setzst
457: lesesc:  move.l d3,d0
458:          bsr setzadr              ; Adresse setzen
459:          bsr srd                  ; Datenwort reinschieben
460:          move.b d0,(a2)+
461:          addq.l #1,d3
462:          cmp.l d3,d4
463:          bcc.s lesesc
464:          bsr aussch
465:          moveq #0,d0
466:          rts
467:
468:  ;          Macht Leertest des Speichers
469:  ;       Parameter : 0/EPROMstart,4/EPROMende
470: ltest:   move.l 0(a5),d3          ; EPROMstart
471:          move.l 4(a5),d4          ; EPROMende
472:          bsr einsch               ; Speicher aktivieren
473:          move.l typzeig(a6),a0
474:          add.w #read,a0
475:          bsr setzst
476: ltsc:    move.l d3,d0
477:          bsr setzadr              ; Adresse setzen
478:          bsr srd                  ; Datenwort reinschieben
479:          cmp.b #$ff,d0            ; Leertest
480:          bne.s ltf
481:          addq.l #1,d3
482:          cmp.l d3,d4
483:          bcc.s ltsc
484:          bsr aussch
485:          move.l d3,d0
486:          subq.l #1,d0
487:          rts
488: ltf:     bsr aussch
489:          move.l d3,d0
490:          subq.l #1,d0
491:          bset #31,d0
492:          rts
493:
494:  ; Vergleicht den Speicher mit dem des Rechners
495:  ;Parameter : 0/Adresse.l,4/EPROMstart,8/EPROMende
496: vergl:   move.l (a5),a2           ; Speicheradresse
497:          move.l 4(a5),d3          ; EPROMstart
498:          move.l 8(a5),d4          ; EPROMende
499:          bsr einsch               ; Speicher aktivieren
500:          move.l typzeig(a6),a0
501:          add.w #read,a0
502:          bsr setzst
503: vgsc:    move.l d3,d0
504:          bsr setzadr              ; Adresse setzen
505:          bsr srd                  ; Datenwort reinschieben
506:          cmp.b (a2)+,d0
507:          bne.s vgf
508:          addq.l #1,d3
509:          cmp.l d3,d4
510:          bcc.s vgsc
511:          bsr aussch
512:          move.l d3,d0
513:          subq.l #1,d0
514:          rts
515: vgf:     bsr aussch
516:          move.l d3,d0
517:          subq.l #1,d0
518:          bset #31,d0
519:          rts
520:
521:  ; Führt einen Verträglichkeitstest der Daten aus
522:  ;Parameter : 0/Adresse.l,4/EPROMstart,8/EPROMende
523: vtest:   move.l (a5),a2           ; Speicheradresse
524:          move.l 4(a5),d3          ; EPROMstart
525:          move.l 8(a5),d4          ; EPROMende
526:          bsr einsch               ; Speicher aktivieren
527:          move.l typzeig(a6),a0
528:          add.w #read,a0
529:          bsr setzst
530: vtsc:    move.l d3,d0
531:          bsr setzadr              ; Adresse setzen
532:          bsr srd                  ; Datenwort reinschieben
533:          not.b d0                 ; invertieren
534:          and.b (a0)+,d0
535:          bne.s vtf
536:          addq.l #1,d3
537:          cmp.l d3,d4
538:          bcc.s vtsc
539:          bsr aussch
540:          move.l d3,d0
541:          subq.l #1,d0
542:          rts
543: vtf:     bsr aussch
544:          move.l d3,d0
545:          subq.l #1,d0
546:          bset #31,d0
547:          rts
548:
549:  ;                  Programmierroutine
550: ipro:    move.l (a5),a3           ; Speicheradresse
551:          move.l 4(a5),d3          ; EPROMstartadresse
552:          move.l 8(a5),d4          ; EPROMendadresse
553:          move.w 12(a5),d5         ; $ff-Flagge
554:          bsr einsch               ; Speicher aktivieren
555:          move.l typzeig(a6),a4
556: progby:
557:          move.l d3,d0
558:          bsr setzadr              ; Adresse
```

```
559:         move.b (a3),d0
560:         tst.w d5                ; $FF überspringen ??
561:         bne.s uff
562:         cmpi.b #$ff,d0
563:         beq ladr
564: uff:    bsr src                 ; An das EPROM
565:         clr.w d2                ; Versuchszähler
566: ippuls:
567:         lea pr_inh(a4),a0
568:         bsr setzst
569:         moveq #1,d0
570:         bsr setzead
571:         lea program(a4),a0
572:         bsr setzst              ; Programmierpuls
573:         move.w i_zeit(a4),d0
574:         bsr zeit                ; Verzögerung
575:         lea pr_inh(a4),a0
576:         bsr setzst              ; aus
577:         addq.w #1,d2            ; Versuchszähler
578:         cmp.w trials(a4),d2     ; Vergleichen
579:         beq.s ipuls
580:         moveq #0,d0
581:         bsr setzead
582:         lea pr_ver(a4),a0       ; Daten vergleichen
583:         bsr setzst
584:         bsr srd
585:         cmp.b (a3),d0
586:         bne.s ippuls            ; nochmal
587: ipuls:  tst.w m_zeit(a4)        ; Multiplikationsfaktor
588:         beq.s nichts
589:         lea pr_inh(a4),a0
590:         bsr setzst
591:         moveq #1,d0
592:         bsr setzead
593:         lea program(a4),a0      ; 2. Programmierimpuls
594:         bsr setzst
595:         move.w d2,d0
596:         mulu m_zeit(a4),d0      ; Multiplikationsfaktor
597:         bsr zeit
598:         lea pr_inh(a4),a0
599:         bsr setzst
600: nichts: cmp.w trials(a4),d2     ; Maximalzeit?
601:         bne.s ladr
602:         moveq #0,d0
603:         bsr setzead             ; Daten einlesen
604:         lea pr_ver(a4),a0
605:         bsr setzst
606:         bsr srd
607:         cmp.b (a3),d0
608:         bne pfeh
609: ladr:
610:         addq.l #1,d3            ; EPROMadresse
611:         addq.l #1,a3            ; Speicheradresse
612:         cmp.l d3,d4
613:         bcc progby              ; Bis zur EPROMendadresse
614:         bsr aussch              ; Speicher abschalten
615:         move.l d3,d0
616:         subq.l #1,d0
617:         rts
618: pfeh:   bsr aussch
619:         move.l d3,d0
620:         subq.l #1,d0
621:         bset #31,d0
622:         rts
623:
624: waltp:  move.l (a5),typzeig(a6) ; Typenzeiger setzen
625:         bsr einsch
626:         bra aussch
627:
628: ; Prüft die Treiberfähigkeit des eingesetzten Speichers
629: tstest: bsr einsch              ; Aktivieren
630:         move.l typzeig(a6),a4
631:         clr.l d0
632:         bsr setzadr
633:         lea read(a4),a0         ; Lesestatus
634:         bsr tstbd
635:         cmp.b d3,d4
636:         bne.s fehl
637:         lea odis(a4),a0
638:         bsr tstbd
639:         cmp.b d3,d4
640:         beq.s fehl
641:         bsr aussch
642:         moveq #0,d0
643:         rts
644: fehl:
645:         bsr aussch
646:         moveq #-1,d0
647:         rts
648:
649:
650: tstbd:  move.l a0,-(sp)
651:         bclr #1,(a0)
652:         bsr setzst
653:         bsr srd
654:         btst #7,d0
655:         seq d3
656:         move.l (sp)+,a0
657:         bset #1,(a0)
658:         bsr setzst
659:         bsr srd
660:         btst #7,d0
661:         seq d4
662:         rts
663:
664:
665: init    clr.l letzta10(a6)
666:         clr.l adrsra(a6)
667:         clr.l letzt_a(a6)
668:         clr.w letzt_b(a6)
669:         lea inaktiv(a6),a0
670:         bsr setzst
671:         clr.l d0
672:         bra setzadr
673:
674: inaktiv dc.b 0
675:         dc.b 0
676:
677: stopikb dc.b $13
678: startikb dc.b $11
679:
680:         align.w
681:
682:                 globale Variablen
683:
684: parbas  dc.l $23334440
685: ergebnis dc.l $23443333
686: alt_ssp dc.l $23334440
687: alt_sr  dc.l $0
688: rwstat  dc.w $2323
689: aseite  dc.w 0
690:
691: istadr  dc.l $44444444
692: typzeig dc.l $44444444
693: adrsra  dc.w 0
694: adrsrb  dc.w 0
695: letzt_a dc.w 0
696: letzt_a2 dc.w 0
697: letzt_b dc.w 0
698:
699: letzta10 dc.l 1
700:
701:         end
```

Listing 2:
Die Typentabelle 'TYPTAB.Q' der Juniorsoft-Funktionen für Profimat. Sie müssen ebenfalls ohne Programmkopf assembliert werden.

```
 1:  ; Typentabelle zu JUNIOR.Q
 2:  ; Typen 2716 bis 27011
 3:
 4:  ; Typ 2716, Index 0
 5:
 6:    dc.b "2716  EPROM  2Kb 25V          " ; Beschreibung
 7:    dc.l 2048      ; Kapazität in Bytes
 8:    dc.l 0         ; Größe einer Seite
 9:    dc.w 0         ; Programmieralgorithmus
10:    dc.w 500       ; i_zeit
11:    dc.w 00        ; m_zeit
12:    dc.w 2         ; trials
13:  ;          AAAAAAAA  BBBBBBBB
14:  ;          76543210  76543210
15:
16:    dc.b %00001000,%01000011      ; Standby
17:    dc.b %00001010,%01000000      ; Read
18:    dc.b %00001010,%01000010      ; Odis
19:    dc.b %10001010,%01000011      ; Program
20:    dc.b %10001010,%01000000      ; Pr_ver
21:    dc.b %10001010,%01000010      ; Pr_inh
22:    dc.b %00000000,%00000000      ; Page_wr
23:    dc.b 0                        ; Manufacturer Code
24:    dc.b 0                        ; Device Code
25:    dc.b %00000000                ; Tsmask
26:    dc.b 0                        ; Frei 3
27:    dc.b %00000000,%00000000      ; Frei 1
28:    dc.b %00000000,%00000000      ; Frei 2
29:
30:    dc.b %00000000,%00001000      ; Tabelle A10
31:    dc.b %00000000,%00000000      ; A11
32:    dc.b %00000000,%00000000      ; A12
33:    dc.b %00000000,%00000000      ; A13
34:    dc.b %00000000,%00000000      ; A14
35:    dc.b %00000000,%00000000      ; A15
```

```
 36:    dc.b %00000000,%00000000        ; A16
 37:
 38: ; Typ 2732, Index 1
 39:
 40:    dc.b "2732  EPROM  4Kb 25V        " ; Beschreibung
 41:    dc.l 4096       ; Kapazität in Bytes
 42:    dc.l 0          ; Größe einer Seite
 43:    dc.w 0          ; Programmieralgorithmus
 44:    dc.w 500        ; i_zeit
 45:    dc.w 0          ; m_zeit
 46:    dc.w 2          ; trials
 47: ;          AAAAAAAA  BBBBBBBB
 48: ;          76543210  76543210
 49:
 50:    dc.b %00000000,%01000011        ; Standby
 51:    dc.b %00000010,%01000000        ; Read
 52:    dc.b %00000010,%01000010        ; Odis
 53:    dc.b %10000010,%01000110        ; Program
 54:    dc.b %10000010,%01000000        ; Pr_ver
 55:    dc.b %10000010,%01000010        ; Pr_inh
 56:    dc.b %00000000,%00000000        ; Page_wr
 57:    dc.b 0                          ; Manufacturer Code
 58:    dc.b 0                          ; Device Code
 59:    dc.b %00000000                  ; Tsmask
 60:    dc.b 0                          ; Frei 3
 61:    dc.b %00000000,%00000000        ; Frei 1
 62:    dc.b %00000000,%00000000        ; Frei 2
 63:
 64:    dc.b %00000000,%00001000        ; Tabelle A10
 65:    dc.b %00000000,%00010000        ; A11
 66:    dc.b %00000000,%00000000        ; A12
 67:    dc.b %00000000,%00000000        ; A13
 68:    dc.b %00000000,%00000000        ; A14
 69:    dc.b %00000000,%00000000        ; A15
 70:    dc.b %00000000,%00000000        ; A16
 71:
 72: ; Typ 2732 A, Index 2
 73:
 74:    dc.b "2732A EPROM  4Kb 21V        " ; Beschreibung
 75:    dc.l 4096       ; Kapazität in Bytes
 76:    dc.l 0          ; Größe einer Seite
 77:    dc.w 0          ; Programmieralgorithmus
 78:    dc.w 500        ; i_zeit
 79:    dc.w 0          ; m_zeit
 80:    dc.w 2          ; trials
 81: ;          AAAAAAAA  BBBBBBBB
 82: ;          76543210  76543210
 83:
 84:    dc.b %00000000,%01000011        ; Standby
 85:    dc.b %00000010,%01000000        ; Read
 86:    dc.b %00000010,%01000010        ; Odis
 87:    dc.b %01000010,%01000110        ; Program
 88:    dc.b %01000010,%01000000        ; Pr_ver
 89:    dc.b %01000010,%01000010        ; Pr_inh
 90:    dc.b %00000000,%00000000        ; Page_wr
 91:    dc.b 0                          ; Manufacturer Code
 92:    dc.b 0                          ; Device Code
 93:    dc.b %00000000                  ; Tsmask
 94:    dc.b 0                          ; Frei 3
 95:    dc.b %00000000,%00000000        ; Frei 1
 96:    dc.b %00000000,%00000000        ; Frei 2
 97:
 98:    dc.b %00000000,%00001000        ; Tabelle A10
 99:    dc.b %00000000,%00010000        ; A11
100:    dc.b %00000000,%00000000        ; A12
101:    dc.b %00000000,%00000000        ; A13
102:    dc.b %00000000,%00000000        ; A14
103:    dc.b %00000000,%00000000        ; A15
104:    dc.b %00000000,%00000000        ; A16
105:
106: ; Typ 2764, Index 3
107:
108:    dc.b "2764  EPROM  8Kb 21V        " ; Beschreibung
109:    dc.l 8192       ; Kapazität in Bytes
110:    dc.l 0          ; Größe einer Seite
111:    dc.w 0          ; Programmieralgorithmus
112:    dc.w 10         ; i_zeit
113:    dc.w 40         ; m_zeit
114:    dc.w 15         ; trials
115: ;          AAAAAAAA  BBBBBBBB
116: ;          76543210  76543210
117:
118:    dc.b %00010010,%10000011        ; Standby
119:    dc.b %00010010,%10000000        ; Read
120:    dc.b %00010010,%10000001        ; Odis
121:    dc.b %01010010,%00000010        ; Program
122:    dc.b %01010010,%10000000        ; Pr_ver
123:    dc.b %01010010,%10000010        ; Pr_inh
124:    dc.b %00000000,%00000000        ; Page_wr
125:    dc.b 0                          ; Manufacturer Code
126:    dc.b 0                          ; Device Code
127:    dc.b %00000000                  ; Tsmask
128:    dc.b 0                          ; Frei 3
129:    dc.b %00000000,%00000000        ; Frei 1
130:    dc.b %00000000,%00000000        ; Frei 2
131:
132:    dc.b %00000000,%00001000        ; Tabelle A10
133:    dc.b %00000000,%00010000        ; A11
134:    dc.b %00000000,%00100000        ; A12
135:    dc.b %00000000,%00000000        ; A13
136:    dc.b %00000000,%00000000        ; A14
137:    dc.b %00000000,%00000000        ; A15
138:    dc.b %00000000,%00000000        ; A16
139:
140: ; Typ 2764A, Index 4
141:
142:    dc.b "2764A EPROM  8Kb 12.5V      " ; Beschreibung
143:    dc.l 8192       ; Kapazität in Bytes
144:    dc.l 0          ; Größe einer Seite
145:    dc.w 0          ; Programmieralgorithmus
146:    dc.w 10         ; i_zeit
147:    dc.w 40         ; m_zeit
148:    dc.w 15         ; trials
149: ;          AAAAAAAA  BBBBBBBB
150: ;          76543210  76543210
151:
152:    dc.b %00010010,%10000011        ; Standby
153:    dc.b %00010010,%10000000        ; Read
154:    dc.b %00010010,%10000001        ; Odis
155:    dc.b %00110010,%00000010        ; Program
156:    dc.b %00110010,%10000000        ; Pr_ver
157:    dc.b %00110010,%10000010        ; Pr_inh
158:    dc.b %00000000,%00000000        ; Page_wr
159:    dc.b 0                          ; Manufacturer Code
160:    dc.b 0                          ; Device Code
161:    dc.b %00000000                  ; Tsmask
162:    dc.b 0                          ; Frei 3
163:    dc.b %00000000,%00000000        ; Frei 1
164:    dc.b %00000000,%00000000        ; Frei 2
165:
166:    dc.b %00000000,%00001000        ; Tabelle A10
167:    dc.b %00000000,%00010000        ; A11
168:    dc.b %00000000,%00100000        ; A12
169:    dc.b %00000000,%00000000        ; A13
170:    dc.b %00000000,%00000000        ; A14
171:    dc.b %00000000,%00000000        ; A15
172:    dc.b %00000000,%00000000        ; A16
173:
174: ; Typ 27128, Index 5
175:
176:    dc.b "27128 EPROM 16Kb 21V        " ; Beschreibung
177:    dc.l 16384      ; Kapazität in Bytes
178:    dc.l 0          ; Größe einer Seite
179:    dc.w 0          ; Programmieralgorithmus
180:    dc.w 10         ; i_zeit
181:    dc.w 40         ; m_zeit
182:    dc.w 15         ; trials
183: ;          AAAAAAAA  BBBBBBBB
184: ;          76543210  76543210
185:
186:    dc.b %00010000,%10000011        ; Standby
187:    dc.b %00010010,%10000000        ; Read
188:    dc.b %00010010,%10000010        ; Odis
189:    dc.b %01010010,%00000010        ; Program
190:    dc.b %01010010,%10000000        ; Pr_ver
191:    dc.b %01010010,%10000010        ; Pr_inh
192:    dc.b %00000000,%00000000        ; Page_wr
193:    dc.b 0                          ; Manufacturer Code
194:    dc.b 0                          ; Device Code
195:    dc.b %00000000                  ; Tsmask
196:    dc.b 0                          ; Frei 3
197:    dc.b %00000000,%00000000        ; Frei 1
198:    dc.b %00000000,%00000000        ; Frei 2
199:
200:    dc.b %00000000,%00001000        ; Tabelle A10
201:    dc.b %00000000,%00010000        ; A11
202:    dc.b %00000000,%00100000        ; A12
203:    dc.b %00000000,%01000000        ; A13
204:    dc.b %00000000,%00000000        ; A14
205:    dc.b %00000000,%00000000        ; A15
206:    dc.b %00000000,%00000000        ; A16
207:
208: ; Typ 27128A, Index 6
209:
210:    dc.b "27128A EPROM 16Kb 12.5V     " ; Beschreibung
211:    dc.l 16384      ; Kapazität in Bytes
212:    dc.l 0          ; Größe einer Seite
213:    dc.w 0          ; Programmieralgorithmus
214:    dc.w 10         ; i_zeit
215:    dc.w 40         ; m_zeit
216:    dc.w 15         ; trials
217: ;          AAAAAAAA  BBBBBBBB
218: ;          76543210  76543210
219:
220:    dc.b %00010010,%10000011        ; Standby
221:    dc.b %00010010,%10000000        ; Read
222:    dc.b %00010010,%10000001        ; Odis
223:    dc.b %00110010,%00000010        ; Program
224:    dc.b %00110010,%10000000        ; Pr_ver
225:    dc.b %00110010,%10000010        ; Pr_inh
```

```
226:    dc.b %00000000,%00000000        ; Page_wr
227:    dc.b 0                          ; Manufacturer Code
228:    dc.b 0                          ; Device Code
229:    dc.b %00000000                  ; Tsmask
230:    dc.b 0                          ; Frei 3
231:    dc.b %00000000,%00000000        ; Frei 1
232:    dc.b %00000000,%00000000        ; Frei 2
233:
234:    dc.b %00000000,%00001000        ; Tabelle A10
235:    dc.b %00000000,%00010000        ; A11
236:    dc.b %00000000,%00100000        ; A12
237:    dc.b %00000000,%01000000        ; A13
238:    dc.b %00000000,%00000000        ; A14
239:    dc.b %00000000,%00000000        ; A15
240:    dc.b %00000000,%00000000        ; A16
241:
242:  ; Typ 27256I, Index 7
243:
244:    dc.b "27256I EPROM 16Kb 12.5V         " ; Beschreibung
245:    dc.l 32768      ; Kapazität in Bytes
246:    dc.l 0          ; Größe einer Seite
247:    dc.w 0          ; Programmieralgorithmus
248:    dc.w 10         ; l_zeit
249:    dc.w 30         ; m_zeit
250:    dc.w 25         ; trials
251:  ;       AAAAAAAA  BBBBBBBB
252:  ;       76543210  76543210
253:
254:    dc.b %00010000,%00000011        ; Standby
255:    dc.b %00010010,%00000000        ; Read
256:    dc.b %00010010,%00000010        ; Odis
257:    dc.b %00110010,%00000010        ; Program
258:    dc.b %00110010,%00000001        ; Pr_ver
259:    dc.b %00110010,%00000011        ; Pr_inh
260:    dc.b %00000000,%00000000        ; Page_wr
261:    dc.b 0                          ; Manufacturer Code
262:    dc.b 0                          ; Device Code
263:    dc.b %00000000                  ; Tsmask
264:    dc.b 0                          ; Frei 3
265:    dc.b %00000000,%00000000        ; Frei 1
266:    dc.b %00000000,%00000000        ; Frei 2
267:
268:    dc.b %00000000,%00001000        ; Tabelle A10
269:    dc.b %00000000,%00010000        ; A11
270:    dc.b %00000000,%00100000        ; A12
271:    dc.b %00000000,%01000000        ; A13
272:    dc.b %00000000,%10000000        ; A14
273:    dc.b %00000000,%00000000        ; A15
274:    dc.b %00000000,%00000000        ; A16
275:
276:  ; Typ 27256, Index 8
277:
278:    dc.b "27256 EPROM 32Kb 21V            " ; Beschreibung
279:    dc.l 32768      ; Kapazität in Bytes
280:    dc.l 0          ; Größe einer Seite
281:    dc.w 0          ; Programmieralgorithmus
282:    dc.w 10         ; l_zeit
283:    dc.w 30         ; m_zeit
284:    dc.w 25         ; trials
285:  ;       AAAAAAAA  BBBBBBBB
286:  ;       76543210  76543210
287:
288:    dc.b %00010000,%00000011        ; Standby
289:    dc.b %00010010,%00000000        ; Read
290:    dc.b %00010010,%00000010        ; Odis
291:    dc.b %01010010,%00000010        ; Program
292:    dc.b %01010010,%00000000        ; Pr_ver
293:    dc.b %01010010,%00000011        ; Pr_inh
294:    dc.b %00000000,%00000000        ; Page_wr
295:    dc.b 0                          ; Manufacturer Code
296:    dc.b 0                          ; Device Code
297:    dc.b %00000000                  ; Tsmask
298:    dc.b 0                          ; Frei 3
299:    dc.b %00000000,%00000000        ; Frei 1
300:    dc.b %00000000,%00000000        ; Frei 2
301:
302:    dc.b %00000000,%00001000        ; Tabelle A10
303:    dc.b %00000000,%00010000        ; A11
304:    dc.b %00000000,%00100000        ; A12
305:    dc.b %00000000,%01000000        ; A13
306:    dc.b %00000000,%10000000        ; A14
307:    dc.b %00000000,%00000000        ; A15
308:    dc.b %00000000,%00000000        ; A16
309:
310:  ; Typ 27512, Index 9
311:
312:    dc.b "27512 EPROM 64Kb 12.5V          " ; Beschreibung
313:    dc.l 65536      ; Kapazität in Bytes
314:    dc.l 0          ; Größe einer Seite
315:    dc.w 0          ; Programmieralgorithmus
316:    dc.w 10         ; l_zeit
317:    dc.w 30         ; m_zeit
318:    dc.w 25         ; trials
319:  ;       AAAAAAAA  BBBBBBBB
320:  ;       76543210  76543210
321:
322:    dc.b %00000000,%00000011        ; Standby
323:    dc.b %00000010,%00000000        ; Read
324:    dc.b %00000010,%00000010        ; Odis
325:    dc.b %00100010,%00000110        ; Program
326:    dc.b %00100010,%00000000        ; Pr_ver
327:    dc.b %00100010,%00000001        ; Pr_inh
328:    dc.b %00000000,%00000000        ; Page_wr
329:    dc.b 0                          ; Manufacturer Code
330:    dc.b 0                          ; Device Code
331:    dc.b %00000000                  ; Tsmask
332:    dc.b 0                          ; Frei 3
333:    dc.b %00000000,%00000000        ; Frei 1
334:    dc.b %00000000,%00000000        ; Frei 2
335:
336:    dc.b %00000000,%00001000        ; Tabelle A10
337:    dc.b %00000000,%00010000        ; A11
338:    dc.b %00000000,%00100000        ; A12
339:    dc.b %00000000,%01000000        ; A13
340:    dc.b %00000000,%10000000        ; A14
341:    dc.b %00000001,%00000000        ; A15
342:    dc.b %00000000,%00000000        ; A16
343:
344:  ; Typ 27513, Index 10
345:
346:    dc.b "27513 EPROM 4*16Kb*8 12.5V      " ; Beschreibung
347:    dc.l 65536      ; Kapazität in Bytes
348:    dc.l 16384      ; Größe einer Seite
349:    dc.w 0          ; Programmieralgorithmus
350:    dc.w 10         ; l_zeit
351:    dc.w 30         ; m_zeit
352:    dc.w 25         ; trials
353:  ;       AAAAAAAA  BBBBBBBB
354:  ;       76543210  76543210
355:
356:    dc.b %00000001,%10000011        ; Standby
357:    dc.b %00000011,%10000000        ; Read
358:    dc.b %00000011,%10000010        ; Odis
359:    dc.b %00100011,%10000110        ; Program
360:    dc.b %00100011,%10000000        ; Pr_ver
361:    dc.b %00100011,%10000011        ; Pr_inh
362:    dc.b %00000000,%00000010        ; Page_wr
363:    dc.b 0                          ; Manufacturer Code
364:    dc.b 0                          ; Device Code
365:    dc.b %00000000                  ; Tsmask
366:    dc.b 0                          ; Frei 3
367:    dc.b %00000000,%00000000        ; Frei 1
368:    dc.b %00000000,%00000000        ; Frei 2
369:
370:    dc.b %00000000,%00001000        ; Tabelle A10
371:    dc.b %00000000,%00010000        ; A11
372:    dc.b %00000000,%00100000        ; A12
373:    dc.b %00000000,%01000000        ; A13
374:    dc.b %00000000,%00000000        ; A14
375:    dc.b %00000000,%00000000        ; A15
376:    dc.b %00000000,%00000000        ; A16
377:
378:  ; Typ 27011, Index 11
379:
380:    dc.b "27011 EPROM 8*16Kb*8 12.5V      " ; Beschreibung
381:    dc.l 131072     ; Kapazität in Bytes
382:    dc.l 16384      ; Größe einer Seite
383:    dc.w 0          ; Programmieralgorithmus
384:    dc.w 10         ; l_zeit
385:    dc.w 30         ; m_zeit
386:    dc.w 25         ; trials
387:  ;       AAAAAAAA  BBBBBBBB
388:  ;       76543210  76543210
389:
390:    dc.b %00010001,%10000011        ; Standby
391:    dc.b %00010011,%10000000        ; Read
392:    dc.b %00010011,%10000010        ; Odis
393:    dc.b %00110011,%00000010        ; Program
394:    dc.b %00110011,%10000000        ; Pr_ver
395:    dc.b %00110011,%10000011        ; Pr_inh
396:    dc.b %00000000,%00000010        ; Page_wr
397:    dc.b 0                          ; Manufacturer Code
398:    dc.b 0                          ; Device Code
399:    dc.b %00000000                  ; Tsmask
400:    dc.b 0                          ; Frei 3
401:    dc.b %00000000,%00000000        ; Frei 1
402:    dc.b %00000000,%00000000        ; Frei 2
403:
404:    dc.b %00000000,%00001000        ; Tabelle A10
405:    dc.b %00000000,%00010000        ; A11
406:    dc.b %00000000,%00100000        ; A12
407:    dc.b %00000000,%01000000        ; A13
408:    dc.b %00000000,%00000000        ; A14
409:    dc.b %00000000,%00000000        ; A15
410:    dc.b %00000000,%00000000        ; A16
411:
412:    end
```

Listing 3:
Das GFA-BASIC-Programm 'JUNIOR.BAS' zum Prüfen und Kalibrieren des Juniorprommers.

```
 1:  ' JUNIOR.BAS - Funktionstest der Hardware und Demon-
 2:  ' stration der Juniorsoft-Routinen aus GFA-BASIC heraus.
 3:  ' Funktionsaufruf:
       Ergebnis% = C:Junior%(Funktionsnr.%,Parameter%,...)
 4:  ' R.Hofmann 1/87
 5:  Junior$=Space$(1000)  ! Programm findet Platz Junior$
 6:  Junior%=Varptr(Junior$)
 7:  Bload ":Juniorpro.b",Junior%   ! Assemblerroutinen
                                      nachladen
 8:  '
 9:  Typ$=Space$(3000)    ! Typentabelle auch in einem String
10:  Typ%=Varptr(Typ$)
11:  Bload "typtab.b",Typ%    ! Typentabelle laden
12:  ' Testschleife
13:  For Test%=1 To 24
14:    Read T$,Sra%,Srb%
15:    Print "Test Nr. ";Test%;": ";T$;
16:    Dummi%=C:Junior%(0,Sra%,0)        ! An SRA ausgeben
17:    Dummi%=C:Junior%(0,Srb%,1)        ! An SRB ausgeben
18:    Input "  Return";Dummi$
19:  Next Test%
20:  ' Testtabelle
21:  Data "Abgleichen der Programmierspannung
           auf 25V an Pin 1",0,0
22:  Data "Prüfen, ob Pin 1 = 21V.",0,0
23:  Data "Prüfen, ob Pin 1 = 12.5V.",0,0
24:  Data "Prüfen, ob Pin 1 = 5V.",0,0
25:  Data "Prüfen, ob Pin 1 = 4.3V.",0,0
26:  Data "Prüfen, ob Pin 1 = 0V.",0,0
27:  Data "Prüfen, ob Pin 2 = 5V.",0,0
28:  Data "Prüfen, ob Pin 2 = 0V.",0,0
29:  Data "Prüfen, ob Pin 20 = 5V.",0,0
30:  Data "Prüfen, ob Pin 20 = 0V.",0,0
31:  Data "Prüfen, ob Pin 21 = 5V.",0,0
32:  Data "Prüfen, ob Pin 21 = 0V.",0,0
33:  Data "Prüfen, ob Pin 22 = 25V.",0,0
34:  Data "Prüfen, ob Pin 22 = 4.3V.",0,0
35:  Data "Prüfen, ob Pin 22 = 0V.",0,0
36:  Data "Prüfen, ob Pin 23 = 25V.",0,0
37:  Data "Prüfen, ob Pin 23 = 4.3V.",0,0
38:  Data "Prüfen, ob Pin 23 = 0V.",0,0
39:  Data "Prüfen, ob Pin 26 = 5V.",0,0
40:  Data "Prüfen, ob Pin 26 = 0V.",0,0
41:  Data "Prüfen, ob Pin 27 = 5V.",0,0
42:  Data "Prüfen, ob Pin 27 = 0V.",0,0
43:  Data "Prüfen, ob die LED leuchtet.",0,0
44:  Data "Prüfen, ob die LED nicht leuchtet.",0,0
```

Listing 4:
Das Beispielprogramm 'BEISPIEL.BAS', welches alle Funktionen benutzt.

```
 1:  ' BEISPIEL.BAS  -  Erläutert alle Juniorsoft-Funktionen
 2:  ' RH 1/87
 3:  Junior$=Space$(1000)      ! In Junior$ werden die
 4:  '                           Juniorsoft-Funktionen geladen
 5:  Junior%=Varptr(Junior$)  ! Junior% enthält die Adresse
                                der Zeichenkette
 6:  Bload "JUNIOR.B",Junior% ! Assemblercode laden
 7:  '
 8:  Typ$=Space$(3000)        ! Wie oben, jedoch für die
 9:  Typ%=Varptr(Typ$)        ! Typentabelle
10:  Bload "TYPTAB.B",Typ%
11:  '
12:  '             Datenspeicher einrichten
13:  '
14:  Daten$=Space$(16385)
15:  Daten%=Varptr(Daten$)
16:  '
17:  '           Zuerst immer den Typ wählen
18:  '
19:  @Typ(5)          ! Typindex 5 -> EPROM 27128, 21 Volt
20:  '
21:  '            Hardware initialisieren
22:  '
23:  Dummi%=C:Junior%(11)
24:  Print "Juniorprommer Demonstration."
25:  '
26:  '       Prüfen,  ob ein EPROM eingesetzt ist
27:  '
28:  @Typda
29:  ' Lesen des EPROM's
30:  Print "Lesen des EPROMs."
31:  Dummi%=C:Junior(4,L:Daten%,L:0,L:Endadr%)
32:  Do
33:    Print "Bitte zu programmierendes Eprom einsetzen."
34:    @Typda                          ! EPROM eingesetzt?
35:    '
36:    '     Leertest und Verträglichkeitstest
37:    '
38:    Print "Leertest des EPROMs."
39:    Exit If C:Junior%(5,L:0,L:Endadr%)>0   ! Leertest
40:    Print "Speicher nicht leer"
41:    Exit If C:Junior%(7,L:Daten%,L:0,L:Endadr%)
                                   ! Verträglichkeitstest
42:    Print "Ein Programmieren ist mit diesen Daten
             nicht möglich."
43:  Loop
44:  '
45:  '                     Programmieren
46:  '
47:  Print "Programmieren ... ";
48:  If C:Junior%(8,L:Daten%,L:0,L:Endadr%,0)>0
                                            ! Programmieren
49:    Print "erfolgreich."
50:  Else
51:    Print "Fehlgeschlagen."
52:    End
53:  Endif
54:  '
55:  '                       Vergleichen
56:  '
57:  Print "Vergleichen ... ";
58:  If C:Junior%(6,L:Daten%,L:0,L:Endadr%,0)>0
                                             ! Vergleichen
59:    Print "erfolgreich."
60:  Else
61:    Print "fehlerhaft."
62:  Endif
63:  End
64:  '
65:  '                    Unterroutinen
66:  '
67:  Procedure Typ(Index%)
68:    Dummi%=C:Junior%(10,L:Typ%+Index%*82)
                                        ! Typ einstellen
69:    Kapazi%=Lpeek(Typ%+Index%*82+30)
                                  ! Kapazität in der Tabelle
70:    Endadr%=Kapazi%-1         ! Resultierende Endadresse
71:  Return
72:  '
73:  '          Prüft, ob Speicher eingesetzt ist
74:  '
75:  Procedure Typda
76:    Do
77:      Exit If C:Junior%(9)=0
78:      Print "Speicher einsetzen"
79:      Do
80:        Exit If Inkey$=Chr$(13)
81:      Loop
82:    Loop
83:  Return
```

Listing 5: 'JUNIOR.Q' als BASIC-Datas

```
10 ' JUNIOR.B Lader (Zeilennummern nicht notwendig)
11 Filename$= "JUNIOR.B"
12 OPEN "O",1,Filename$
13 REPEAT
14   C=C+1
15   READ Wert
16   IF Wert=-1 THEN
17     END
18   ENDIF
19   IF (C MOD 11)=0 THEN
20     IF Wert<>(Summe AND 255) THEN   ,
21       PRINT"Fehler in Datazeile "C/11
22       END
23     ENDIF
24     Summe=0
26   ELSE
27     PRINT #1, CHR$(Wert);
28     Summe=Summe+(Wert+C MOD 11)^2
29   ENDIF
30 UNTIL Wert=-1
31 CLOSE(1)
100 DATA 72,231,127,254,77,250,255,250,75,239,23
101 DATA 0,62,45,77,6,114,66,167,63,60,49
102 DATA 0,32,78,65,92,143,45,64,6,122,250
```

Das Buch dazu



ca. 300 Seiten, incl. Diskette

GFA-Buch ATARI ST

Incl. Diskette mit Beispiel-Programmen

GFA BASIC Version 3.0

- Befehle
- Anwendungen
- Möglichkeiten

Gottfried P. Engels
Markus C. Görgens

DM 59,–

...Anruf genügt.

GFA-CLUB
GFA-ST/PC-Software bitte Info anfordern

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 02 11/58 80 11



```
103 DATA 65,238,6,112,47,8,66,103,63,60,143
104 DATA 0,25,78,78,80,143,64,238,6,126,231
105 DATA 48,60,0,100,97,0,3,112,0,124,143
106 DATA 7,0,97,0,0,44,47,0,70,238,134
107 DATA 6,126,65,238,6,113,47,8,66,103,13
108 DATA 63,60,0,25,78,78,80,143,47,46,241
109 DATA 6,122,63,60,0,32,78,65,92,143,86
110 DATA 32,31,76,223.127,254.78.117.48.45.16
111 DATA 255,254,74,64,107,246,176,124,0,11,52
112 DATA 98,240,229,72,32,123,0,6,209,206,16
113 DATA 78,208,0,0,1,36,0,0,1,190,219
114 DATA 0,0,2,48,0,0,2,114,0,0,21
115 DATA 4,2,0,0,4,50,0,0,4,116,241
116 DATA 0,0,4,186,0,0,5,2,0,0,32
117 DATA 5,240,0,0,5,228,0,0,6,80,63
118 DATA 72,231,160,0,114,2,128,46,6,142,2
119 DATA 63,0,97,0,0,110,48,23,18,0,156
120 DATA 192,60,0,224,20,46,6,148,196,60,165
121 DATA 0,224,180,0,103,18,85,194,48,60,127
122 DATA 3,232,74,2,102,4,48,60,31,64,227
123 DATA 97,0,2,192,194,60,0,1,16,46,153
124 DATA 6,148,192,60,0,1,178,0,103,6,217
125 DATA 112,50,97,0,2,170,48,31,29,64,140
126 DATA 6,148,76,223,0,5,29,64,6,146,146
127 DATA 78,117,114,4,29,64,6,150,128,46,209
128 DATA 6,144,10,0,0,64,96,16,114,8,121
129 DATA 96,12,48,21,114,1,227,73,83,109,135
130 DATA 0,2,102,248,72,231,48,128,65,248,61
131 DATA 136,2,17,124,0,15,255,254,54,46,204
132 DATA 6,130,52,3,227,16,226,18,16,130,29
133 DATA 132,1,16,130,52,3,227,16,226,18,158
134 DATA 16,130,132,1,16,130,52,3,227,16,186
135 DATA 226,18,16,130,132,1,16,130,52,3,143
136 DATA 227,16,226,18,16,130,132,1,16,130,213
137 DATA 52,3,227,16,226,18,16,130,132,1,142
138 DATA 16,130,52,3,227,16,226,18,16,130,13
139 DATA 132,1,16,130,52,3,227,16,226,18,158
140 DATA 16,130,132,1,16,130,52,3,227,16,186
141 DATA 226,18,16,130,132,1,16,130,17,124,37
142 DATA 0,14,255,254,20,40,255,254,132,60,211
143 DATA 0,32,16,130,196,60,0,223,16,130,138
144 DATA 76,223,1,12,78,117,72,231,24,96,163
145 DATA 67,248,136,2,19,124,0,14,255,254,170
146 DATA 22,17,134,60,0,32,18,131,69,248,58
147 DATA 250,0,198,60,0,223,18,131,118,32,175
148 DATA 120,0,112,0,19,124,0,15,255,254,244
149 DATA 228,210,18,131,209,64,18,132,228,210,213
150 DATA 18,131,209,64,18,132,228,210,18,131,118
151 DATA 209,64,18,132,228,210,18,131,209,64,132
152 DATA 18,132,228,210,18,131,209,64,18,132,97
153 DATA 228,210,18,131,209,64,18,132,228,210,213
154 DATA 18,131,209,64,18,132,228,210,18,131,118
155 DATA 209,64,70,0,76,223,6,24,78,117,222
156 DATA 74,85,103,0,0,32,17,252,0,15,77
157 DATA 136,0,48,46,6,130,226,72,0,0,193
158 DATA 0,16,17,192,136,2,192,60,0,239,17
159 DATA 17,192,136,2,78,117,17,252,0,15,249
160 DATA 136,0,48,46,6,130,226,72,128,60,129
161 DATA 0,64,17,192,136,2,192,60,0,191,113
162 DATA 17,192,136,2,78,117,48,21,192,124,246
163 DATA 0,1,17,252,0,15,136,0,17,192,149
164 DATA 136,2,227,72,61,64,6,130,78,117,92
165 DATA 72,231,120,192,36,0,34,110,6,138,38
166 DATA 180,174,6,134,103,118,98,10,66,174,220
167 DATA 6,134,97,178,96,0,0,8,82,174,172
168 DATA 6,134,97,138,34,46,6,134,224,137,173
169 DATA 228,137,178,174,6,152,103,218,74,169,230
170 DATA 0,34,103,22,38,46,6,134,134,233,89
171 DATA 0,36,182,110,6,132,103,8,61,67,136
172 DATA 6,132,97,0,0,66,45,65,6,152,148
173 DATA 32,73,208,252,0,68,112,6,66,67,215
174 DATA 66,68,226,73,100,6,134,16,136,40,30
175 DATA 0,1,84,136,81,200,255,242,29,67,232
176 DATA 6,142,29,68,6,144,16,46,6,146,52
177 DATA 97,0,253,176,16,46,6,150,97,0,214
178 DATA 254,4,96,132,76,223,3,30,78,117,194
179 DATA 72,231,240,192,34,110,6,138,48,46,250
180 DATA 6,132,97,0,253,252,112,1,97,0,103
181 DATA 255,78,20,46,6,146,22,46,6,150,28
182 DATA 16,2,2,0,0,231,97,0,253,120,132
183 DATA 48,60,0,100,97,0,0,100,16,41,209
184 DATA 0,51,97,0,253,196,16,41,0,59,222
185 DATA 97,0,253,188,16,41,0,51,97,0,244
186 DATA 253,180,16,2,97,0,253,82,16,3,139
187 DATA 97,0,253,168,48,60,0,20,97,0,158
188 DATA 0,56,112,0,97,0,255,2,76,223,120
189 DATA 3,15,78,117,16,2,97,0,253,50,176
190 DATA 16,3,96,0,253,136,47,8,16.24.180
191 DATA 176,46,6,146,103,4,97,0,253,30,110
192 DATA 16,24,176,46,6,150,103,4,97,0,87
193 DATA 253,110,32,95,78,117,103,14,192,252,53
194 DATA 0,40,144,188,0,0,0,38,83,128,116
195 DATA 102,252,78,117,66,174,6,142,66,174,78
196 DATA 6,152,66,110,6,146,66,110,6,148,133
197 DATA 66,110,6,150,66,110,6,132,112,0,141
198 DATA 97,0,254,162,32,110,6,138,208,252,24
199 DATA 0,50,97,168,97,0,255,50,66,128,158
200 DATA 96,0,254,160,112,0,97,0,254,136,104
201 DATA 32,110,6,138,208,252,0,46,97,142,76
202 DATA 66,128,96,0,254,138,36,85,38,45,99
203 DATA 0,4,40,45,0,8,97,172,32,110,61
204 DATA 6,138,208,252,0,48,97,0,255,114,247
205 DATA 32,3,97,0,254,108,97,0,253,156,143
206 DATA 20,192,82,131,184,131,100,238,97,190,66
207 DATA 112,0,78,117,38,45,0,0,40,45,188
208 DATA 0,4,97,0,255,126,32,110,6,138,43
209 DATA 208,252,0,48,97,0,255,66,32,3,200
210 DATA 97,0,254,60,97,0,253,108,176,60,198
211 DATA 0,255,102,14,82,131,184,131,100,234,44
212 DATA 97,138,32,3,83,128,78,117,97,130,40
213 DATA 32,3,83,128,8,192,0,31,78,117,207
214 DATA 36,85,38,45,0,4,40,45,0,8,136
215 DATA 97,0,255,58,32,110,6,138,208,252,123
216 DATA 0,48,97,0,254,254,32,3,97,0,176
217 DATA 253,248,97,0,253,40,176,26,102,16,170
218 DATA 82,131,184,131,100,236,97,0,255,72,165
219 DATA 32,3,83,128,78,117,97,0,255,62,80
220 DATA 32,3,83,128,8,192,0,31,78,117,207
221 DATA 36,85,38,45,0,4,40,45,0,8,136
222 DATA 97,0,254,244,32,110,6,138,208,252,180
223 DATA 0,48,97,0,254,184,32,3,97,0,164
224 DATA 253,178,97,0,252,226,70,0,192,24,179
225 DATA 102,16,82,131,184,131,100,234,97,0,190
226 DATA 255,0,32,3,83,128,78,117,97,0,236
227 DATA 254,246,32,3,83,128,8,192,0,31,162
228 DATA 78,117,38,85,38,45,0,4,40,45,137
229 DATA 0,8,58,45,0,12,97,0,254,168,33
230 DATA 40,110,6,138,32,3,97,0,253,110,52
231 DATA 16,19,74,69,102,8,12,0,0,255,156
232 DATA 103,0,0,152,97,0,251,242,66,66,214
233 DATA 65,236,0,56,97,0,254,82,112,1,144
234 DATA 97,0,253,58,65,236,0,52,97,0,253
235 DATA 254,68,48,44,0,40,97,0,254,90,108
236 DATA 65,236,0,56,97,0,254,52,82,66,59
237 DATA 180,108,0,44,103,22,112,0,97,0,191
238 DATA 253,20,65,236,0,54,97,0,254,30,18
239 DATA 97,0,252,78,176,19,102,188,74,108,253
240 DATA 0,42,103,40,65,236,0,56,97,0,102
241 DATA 254,8,112,1,97,0,252,240,65,236,0
242 DATA 0,52,97,0,253,250,48,2,192,236,243
243 DATA 0,42,97,0,254,14,65,236,0,56,111
244 DATA 97,0,253,232,180,108,0,44,102,24,243
245 DATA 112,0,97,0,252,202,65,236,0,54,39
246 DATA 97,0,253,212,97,0,252,4,176,19,35
247 DATA 102,0,0,22,82,131,82,139,184,131,136
248 DATA 100,0,255,78,97,0,254,30,32,3,228
249 DATA 83,128,78,117,97,0,254,20,32,3,145
250 DATA 83,128,8,192,0,31,78,117,45,85,118
251 DATA 6,138,97,0,253,208,96,0,253,254,222
252 DATA 97,0,253,200,40,110,6,138,66,128,219
253 DATA 97,0,252,142,65,236,0,48,97,0,110
254 DATA 0,34,184,3,102,20,65,236,0,50,105
255 DATA 97,0,0,22,184,3,103,8,97,0,155
256 DATA 253,212,112,0,78,117,97,0,253,204,43
257 DATA 112,255,78,117,47,8,8,144,0,1,243
258 DATA 97,0,253,92,97,0,251,140,8,0,105
259 DATA 0,7,87,195,32,95,8,208,0,1,196
260 DATA 97,0,253,72,97,0,251,120,8,0,105
261 DATA 0,7,87,196,78,117,66,174,6,152,210
262 DATA 66,174,6,142,66,174,6,146,66,110,229
263 DATA 6,150,65,238,6,110,97,0,253,36,54
264 DATA 66,128,96,0,252,30,0,0,19,17,81
265 DATA 35,51,68,64,35,68,51,51,35,51,68
266 DATA 68,64,0,0,0,0,35,35,0,0,197
267 DATA 68,68,68,68,68,68,68,68,0,0,33
268 DATA 0,0,0,0,0,0,0,0,0,0,129
269 DATA 0,1,0,0,0,0,0,0,0,0,134
9999 Data -1
```

Listing 6: 'TYPTAB.Q' als BASIC-Datas

```
10 ' TYPTAB.B Lader (Zeilennummern nicht notwendig)
11 Filename$= "TYPTAB.B"
12 OPEN "O",1,Filename$
13 REPEAT
14   C=C+1
15   READ Wert
16   IF Wert=-1 THEN
17     END
18   ENDIF
19   IF (C MOD 11)=0 THEN
20      IF Wert<>(Summe AND 255) THEN
21        PRINT"Fehler in Datazeile "C/11
22        END
23      ENDIF
```

Reinhard Schuster Computer

OBERE MÜNSTERSTR. 33–35 · TEL. (0 23 05) 37 70 · BTX 023053770 · 4620 CASTROP-RAUXEL



Vertragshändler

ATARI System-Fachhändler

star Vertragshändler – der ComputerDrucker

Tandon Computer Vertragshändler

Commodore Vertrags-Werkstatt

ATARI-ST-SOFTWARE

10Th Frame 66.60
2218 Baker Street 47.90
3D Galaxy 54.10
500 CC Grand Prix 58.80
Addicta Ball 47.90
Advanced Art Studio 76.60
Airball 71.20
Airball Construction Set 47.90
Altair 54.10
Alternate Reality City 76.60
Amazon 54.10
Annals of Rome 74.30
Arena 88.30
Arena/Braticass 91.00
Arkanoid 38.60

Baseball 71.20
Championship Football 76.60
Championship Wrestling 71.20
Checkmate 33.60
Chessmaster 30.90
Chopper X 27.80
Clever & Smart 58.80
Colonial Conquest 85.20
Crafton & Xunk 62.30
Crash Garet 62.30
Crazy Cars 55.70
Creator 62.30
Crystal Castles 47.90
Cutthroats 91.00
Dark Castle 71.20
Deathtrike 33.60

BRANDNEU:
MEWILLO ST
Deutsches Adventure **65,—**

Gold Runner 71.20
Golden Path 57.20
Guild of Thieves 71.20
Gunship 74.30
Hacker 61.90
Hacker 2 71.20
Hardball 66.60
Hellowoon 61.90
Hitchhikers Guide to Galaxy 88.30
Hollywood Hi Jinx 88.30
Hollywood Poker 47.90
Hot Ball 71.20
Hunt for Red October 75.70
Impact 43.30
Indiana Jones 52.60
Infidel 91.00
International Karate 62.30
Into the Eagles Nest 58.80
Isnogud 62.30
Jagd auf Roter Oktober 74.30
Jewels of Darkness 62.30
Jinxter 74.30
Joe Blade 33.60
Jump Jet 43.30
Jupiter Probe 43.30
Karate Kid 2 66.60
Karate Master 42.20
Karting Grand Prix 29.30

Liberator 42.20
Little Computer People 105.40
Livingstone 47.90
Luky Luke
Nitroglyzerin 57.20
Lurking Horror 88.30
MacAdam Bumper 62.30
Marble Madness 83.60
Mean 18 Golf 91.00
Mercenary 66.60
Metrocross 66.60
Metropolis 35.50
MGT 71.20
Microleague Wrestling 58.80
Mindshadow 61.90
Mision One Droid 33.60
Mission Elevator 58.80
Moebius 74.30
Moonmist 91.00
Mordville Manor 71.20
Mouse Trap 43.30
Music Studio 105.40
Nine Princess in Amber 54.10
Ninja Mission 30.90
Nord & Bert 76.60
Not a Penny More 62.30
Ogre 74.30

Pirates of the Barbery Coast 35.50
Planetfall 91.00
Plundered Hearts 76.60
Plutos 47.90
Police Quest 58.80
Pool/Shuffle Board 47.90
Predator 62.30
Protector 30.90
Psion Chess 76.60
Q Ball 57.20
Quantum Paint Box 62.30
Rampage 47.90
Rana Rama 58.80
Rings of Zilfin 66.60
Roadrunner 66.60
Roadwar 2000 76.60
Roadwar Europe 66.60
Rockford 58.80
Rogue 76.60
Sapiens 58.80
Scenery Disk 2 76.60
Seastalker 91.00
Seconds Out 55.70
Sentinel 58.80
Shanghai 71.20
Shuttle 2 66.60
Sidewalk 58.80
Silicon Dreams 62.30
Sky Blaster 62.30
Sky Flighter 43.30

ST Classics 62.30
ST-Soccer 55.70
Star Raiders 43.30
Starglider 71.20
Startrek 58.80
Starwars 58.80
Stationfall 88.30
Strike 33.60
Strike Force Harrier 71.20
Strip Poker 62.30
Strip Poker 2 43.30
Sub Battle Simulator 66.60
Super Cycle 76.60
Super Huey 62.30
Super Sprint 43.30
Superstar Icehockey 71.20
Suspect 91.00
Swooper 62.30
T.N.T. 52.60
Taipan 51.00

Warzone 30.90
Waterskiing 58.80
Western Games 58.80
Winter Olympiad '88 55.70
Wintergames 71.20
Wishbringer 91.00
Wizball 55.70
Wizzards Crown 76.60
World Series Baseball 52.60
Xenon 58.80
Xevious 66.60
Zork 1 89.80

Anwender-Software

GFA Basic Interpreter 99.00

BRANDNEU:
LEATHER NECKS ST **49,90**

Asphai Trilogy 76.60
Asterix im Morgenland 61.90
Autoduel 58.80
Backlash 52.60
Bad Cat 55.70
Balance of Power 85.20
Barbarian 43.30
Bard's Tale 1 82.80
Baseball Gamestar 76.60
Beyond Zork 76.60
Black Cauldron 71.20
Black Lamp 58.80
Blue War 54.10
Blueberry und das Gespenst 61.90
Bob Moran 62.30
Bob Moran in Middle Age 62.30
Borrowed Time 61.90
Boulderdash Construction Set 71.20
Brataccas 105.40
Brian Clough Football 76.60
Bubble Bobble 58.80
Bubble Ghost 62.30
Bureaucrazy 99.10
Chamonix Challenge 58.80
Championship

Deep Space 105.40
Defender of the Crown 77.40
Deja Vu 66.60
Diablo 62.30
Dizzy Wizzard 55.70
Dungeon Master 74.30
Eco 58.80
Eden Blues 71.20
Electronic Pool 57.20
Enduro Racer 58.80
Epyx (The Worlds Greatest) 77.40
Extensor 62.30
Extravaganza 42.20
Eye 43.30
Fahrenheit 451 54.10
Fire Blaster 30.90
Flight Sim. Scenery Disk 2 57.20
Flight Simulator 2 138.00
Flight Simulator Scenery Disk 7 57.20
Football Fortunes 71.20
Football Manager 43.30
Frost Byte 47.90
Gambler 38.60
Gato 85.20
Gauntlet 66.60
Gnome Ranger 43.30

Telefonische Bestellung:
(02305) 3770
(Tag und Nacht)

Kings Quest 3 76.60
Kings Quest 3er Pack 76.60
Knight Orc 58.80
L'Affaire 76.60
Las Vegas 33.60
Leaderboard Golf 71.20
Leaderbord Tournament 33.60
Leather Goddess of Phobes 82.10
Leisure Suit Larry 62.30
Leviathan 43.30

Outcast 30.90
Pacland 58.80
Passengers on the Wind 62.30
Passengers on the Wind 2 62.30
Perfect Match 33.60
Perry Mason 54.10
Phantasie 2 76.60
Phantasie 3 66.60
Phoenix 58.80
Pinball Factory 66.60

Skyrider 58.80
Slaygon Adventure 55.70
Solomons Key 54.10
Space Ace 58.80
Space Baller 29.30
Space Pilot 47.90
Space Port 58.80
Space Quest 76.60
Space Quest 2 58.80
Space Station 46.40
Spiderman 62.30
Spy Versus Spy 66.60

Tanglewood 54.10
Tass Times 71.20
Tee up Golf 43.30
Terramax 62.30
Terrorpods 66.60
Tetris 54.10
Thai Boxing 43.30
The Pawn 71.20
Time Bandit 91.00
Time Blast 33.60
Tracker 74.30
Trailblazer 63.50
Trasheap 58.80
Trauma 62.30
Trinity 105.40
Trivial Genus Edit. 62.30
Triviar Trove 27.80
Turbo 27.80
Turbo GT 50.80
Turbo ST 33.60
Two on Two Basketball 76.60
Typhon 62.30
Ultima 2 91.00
Ultima 3 71.20
Ultima 4 74.30
Universal Military Sim. 74.30
Vegas Gambler 52.60
Vermeer 77.40

GFA Basic Compiler 99.00
GFA Basic 68881 349.00
GFA Artist 149.00
GFA Draft 298.00
GFA Draft Plus 349.00
GFA Objekt 99.00
GFA Starter 59.00
GFA Vektor 149.00
Lattice C Ver. 3.04 deutsch 298.00
Lisp Interpreter/Compiler MCC 350.00
Logistix 399.00
Makro Assembler GST 129.00
Macro Assembler MCC 159.00
UCSD-P Syst. incl. UCSD Pascal 198.00
Publishing Partner 498.00
Adimens ST 199.00
Aditalk ST 189.00
1st Word Plus/1st Mail 199.00
Star-Comm 79.00
Fleet Street Publisher 348.00
Omikron Basic 179.00
Omikron Compiler 179.00

HARDWARE

Unser Superknüller Akustikkoppler Hitrans 300 P

300 Baud, vollduplex, RS 232 C-Schnittstelle, flexibles Mittelteil, Stromversorgung über Batterie, Akku oder mitgeliefertem Netzteil möglich. Incl. FTZ-Nr. (Postzulassung), Handbuch und Netzteil.

198,—



Kunstlederhauben

260 /520 ST	17.90
520/1040 STF	21.90
Mega ST Keyboard	21.90
Mega ST Keyboard/SM 124	46.90
Mega Keyboard/SM 125	48.90
Floppy 314/354	14.90
Monitor SM 124	27.90
Monitor SM 125	29.80
Monitor SC 1224	32.90

Zweitlaufwerk 3½'', 720 KB mit Netzteil **348,—**

Markendisketten:

3½'' 1D	24.90
3½''-2 D	29.80
Mouse-Pad	19,80
Floppy-Stecker, 14-pol.	7,90
Floppy-Kupplung 14-pol.	7.90
Monitor-Stecker 13-pol.	7,90
Monitor-Kuppl. 13-pol.	7,90

Diskettenbox 3 + 3½'' für 80 3½ Disketten, abschließbar **17.90**



Ladengeschäftszeiten:
Montag–Freitag 9.00 – 13.00 Uhr
15.00 – 18.30 Uhr
Samstag 9.00 – 14.00 Uhr
Langer Samstag 9.00 – 18.00 Uhr

Versand per Nachnahme zuzügl. Versandkosten. Oder Vorkasse auf Psch.-Kto. Nr. 69422-460 PschA Dortmund zuzügl. 5,– DM Versandkosten.
Ausland nur per Vorkasse auf Psch.-Kto. zuzügl. 10,– DM Versandkosten. **Bitte bei allen Bestellungen Computertyp angeben!**

Besuchen Sie unser Ladengeschäft und lassen Sie sich durch unser Fachpersonal beraten. Wir haben laufend günstige Angebote und stark reduzierte Vorführgeräte.

○ Senden Sie mir bitte Ihren Katalog (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme:
○ Incl. kostenlosem Katalog

Vorname, Name

Straße, Hausnummer

PLZ, Ort

Datum, Unterschrift

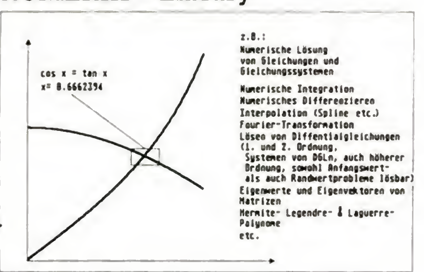
BESTELLSCHEIN

Anz.	Artikel	Preis

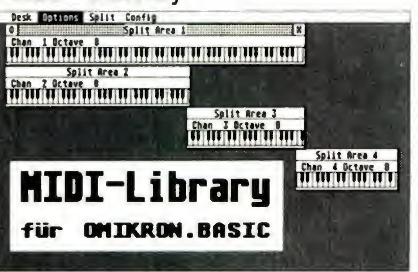
```
24      Summe=0
26    ELSE
27      PRINT #1, CHR$(Wert);
28      Summe=Summe+(Wert+C MOD 11)^2
29    ENDIF
30 UNTIL Wert=-1
31 CLOSE(1)
100 DATA 50,55,49,54,32,32,69,80,82,79,165
101 DATA 77,32,32,50,75,98,32,50,53,86,206
102 DATA 32,32,32,32,32,32,32,32,32,32,65
103 DATA 0,0,8,0,0,0,0,0,0,0,241
104 DATA 1,244,0,0,0,2,8,67,10,64,129
105 DATA 10,66,138,67,138,64,138,66,0,0,34
106 DATA 0,0,0,0,0,0,0,0,0,8,97
107 DATA 0,0,0,0,0,0,0,0,0,0,129
108 DATA 0,0,50,55,51,50,32,32,69,80,176
109 DATA 82,79,77,32,32,52,75,98,32,50,224
110 DATA 53,86,32,32,32,32,32,32,32,32,32
111 DATA 32,32,0,0,16,0,0,0,0,0,225
112 DATA 0,0,1,244,0,0,0,2,0,67,161
113 DATA 2,64,2,66,130,70,130,64,130,66,165
114 DATA 0,0,0,0,0,0,0,0,0,0,129
115 DATA 0,8,0,16,0,0,0,0,0,0,97
116 DATA 0,0,0,0,50,55,51,50,65,32,8
117 DATA 69,80,82,79,77,32,32,52,75,98,147
118 DATA 32,50,49,86,32,32,32,32,32,32,168
119 DATA 32,32,32,32,0,0,16,0,0,0,225
120 DATA 0,0,0,0,1,244,0,0,0,2,184
121 DATA 0,67,2,64,2,66,66,70,66,64,38
122 DATA 66,66,0,0,0,0,0,0,0,0,21
123 DATA 0,0,0,8,0,16,0,0,0,0,193
124 DATA 0,0,0,0,0,0,50,55,54,52,18
125 DATA 32,32,69,80,82,79,77,32,32,56,196
126 DATA 75,98,32,50,49,86,32,32,32,32,231
127 DATA 32,32,32,32,32,32,0,0,32,0,1
128 DATA 0,0,0,0,0,0,0,10,0,40,229
129 DATA 0,15,18,131,18,128,18,129,82,2,236
130 DATA 82,128,82,130,0,0,0,0,0,0,173
131 DATA 0,0,0,0,0,8,0,16,0,32,161
132 DATA 0,0,0,0,0,0,0,0,50,55,230
133 DATA 54,52,65,32,69,80,82,79,77,32,231
134 DATA 32,56,75,98,32,49,50,46,53,86,16
135 DATA 32,32,32,32,32,32,32,32,0,0,129
136 DATA 32,0,0,0,0,0,0,0,0,10,237
137 DATA 0,40,0,15,18,131,18,128,18,129,188
138 DATA 50,2,50,128,50,130,0,0,0,0,121
139 DATA 0,0,0,0,0,0,0,8,0,16,129
140 DATA 0,32,0,0,0,0,0,0,0,0,1
141 DATA 50,55,49,50,56,32,69,80,82,79,21
142 DATA 77,32,49,54,75,98,32,50,49,86,117
143 DATA 32,32,32,32,32,32,32,32,32,32,65
144 DATA 0,0,64,0,0,0,0,0,0,0,1
145 DATA 0,10,0,40,0,15,16,131,18,128,195
146 DATA 18,130,82,2,82,128,82,130,0,0,149
```

```
147 DATA 0,0,0,0,0,0,0,0,0,8,97
148 DATA 0,16,0,32,0,64,0,0,0,0,193
149 DATA 0,0,50,55,49,50,56,65,32,69,127
150 DATA 80,82,79,77,32,49,54,75,98,32,79
151 DATA 49,50,46,53,86,32,32,32,32,32,9
152 DATA 32,32,0,0,64,0,0,0,0,0,193
153 DATA 0,0,0,10,0,40,0,15,18,131,243
154 DATA 18,128,18,129,50,2,50,128,50,130,106
155 DATA 0,0,0,0,0,0,0,0,0,0,129
156 DATA 0,8,0,16,0,32,0,64,0,0,225
157 DATA 0,0,0,0,50,55,50,53,54,73,28
158 DATA 32,69,80,82,79,77,32,49,54,75,148
159 DATA 98,32,49,50,46,53,86,32,32,32,1
160 DATA 32,32,32,32,0,0,128,0,0,0,1
161 DATA 0,0,0,0,0,10,0,30,0,25,38
162 DATA 16,3,18,0,18,2,50,2,50,1,123
163 DATA 50,3,0,0,0,0,0,0,0,0,190
164 DATA 0,0,0,8,0,16,0,32,0,64,193
165 DATA 0,128,0,0,0,0,50,55,50,53,167
166 DATA 54,32,69,80,82,79,77,32,51,50,223
167 DATA 75,98,32,50,49,86,32,32,32,32,231
168 DATA 32,32,32,32,32,32,0,0,128,0,193
169 DATA 0,0,0,0,0,0,0,10,0,30,97
170 DATA 0,25,16,3,18,0,18,2,82,2,103
171 DATA 82,0,82,3,0,0,0,0,0,0,186
172 DATA 0,0,0,0,0,8,0,16,0,32,161
173 DATA 0,64,0,128,0,0,0,0,50,55,230
174 DATA 53,49,50,32,69,80,82,79,77,32,40
175 DATA 54,52,75,98,32,49,50,46,53,86,224
176 DATA 32,32,32,32,32,32,32,32,0,1,150
177 DATA 0,0,0,0,0,0,0,0,0,10,173
178 DATA 0,30,0,25,0,3,2,0,2,2,87
179 DATA 34,6,34,0,34,1,0,0,0,0,186
180 DATA 0,0,0,0,0,0,0,8,0,16,129
181 DATA 0,32,0,64,0,128,1,0,0,0,16
182 DATA 50,55,53,49,51,32,69,80,82,79,17
183 DATA 77,32,52,42,49,54,75,98,42,56,38
184 DATA 32,49,50,46,53,86,32,32,32,32,1
185 DATA 0,1,0,0,0,0,64,0,0,0,6
186 DATA 0,10,0,30,0,25,1,131,3,128,165
187 DATA 3,130,35,134,35,128,35,131,0,2,202
188 DATA 0,0,0,0,0,0,0,0,0,8,97
189 DATA 0,16,0,32,0,64,0,0,0,0,193
190 DATA 0,0,50,55,48,49,49,32,69,80,73
191 DATA 82,79,77,32,56,42,49,54,75,98,135
192 DATA 42,56,32,49,50,46,53,86,32,32,137
193 DATA 32,32,0,2,0,0,0,0,64,0,213
194 DATA 0,0,0,10,0,30,0,25,17,131,186
195 DATA 19,128,19,130,51,2,51,128,51,131,57
196 DATA 0,2,0,0,0,0,0,0,0,0,141
197 DATA 0,8,0,16,0,32,0,64,0,0,225
198 DATA 0,0,0,0,0,0,0,0,0,0,129
9999 Data -1
```

STATISTIK – Library



NUMERIK – Library



MIDI – Library



MASKENEDITOR



D. R. A. W.!



OMIKRON.BASIC: das Entwicklungssystem

„Der OMIKRON.BASIC-Interpreter stellt zusammen mit dem Compiler das beste BASIC-Entwicklungssystem für den Atari ST dar."
(ST Computer 9/87, S. 64)

Der Interpreter

Ein umfangreicher Befehlssatz, ein leistungsstarker Editor und höchste Ausführungsgeschwindigkeit kennzeichnen den OMIKRON.Interpreter.

- 19 Stellen Genauigkeit bei allen Funktionen
- Matrizen-Algebra
- Masken-INPUT, SORT-Befehl (mit deutschen Umlauten)
- Strukturiertes Programmieren: rekursive Prozeduren, mehrzeilige Funktionen
- 99 % MBASIC-kompatibel

OMIKRON.BASIC auf Diskette	**DM 179,–**
OMIKRON.BASIC auf Modul	**DM 229,–**

Der Compiler

Noch mehr Tempo erreichen Ihre in OMIKRON.BASIC geschriebenen Programme durch unseren OMIKRON.Compiler.
Der Compiler erzeugt ein Assemblerprogramm (echte Maschinensprache, keinen P-Code!), das direkt vom Desktop aus angeklickt werden kann. Auch Accessories können Sie mit dem OMIKRON.Compiler erstellen. Durch CUTLIB werden nur diejenigen Ausführungsroutinen eingebunden, die tatsächlich gebraucht werden; die Folge sind außerordentlich kompakte Programme. Die Ablaufgeschwindigkeit der compilierten Programme ist außergewöhnlich. Der Tester in der mc meinte gar: **„Nur Assembler ist schneller"** (mc 2/87, S. 122)

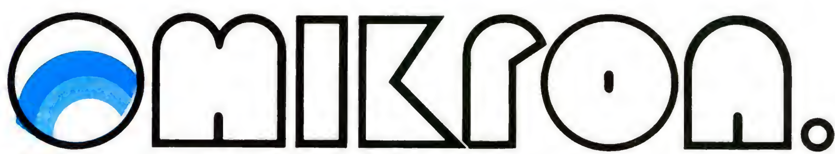
OMIKRON.Compiler	**DM 179,–**
OMIKRON.Compiler für 68881 (vollkompatibel zum normalen Compiler)	**DM 229,–**

Die Libraries

Wer professionell Programme entwickelt, möchte nicht jedesmal das Rad neu erfinden. Mit neuen Libraries (Befehls-Bibliotheken) und Hilfsprogrammen macht OMIKRON.BASIC Ihnen das Leben jetzt noch etwas leichter. Libraries erweitern den Befehlssatz: Sie laden einfach die Library, und die neuen Befehle stehen Ihnen sofort zur Verfügung.
Außer den links abgebildeten Programmen gibt's noch eine Robot-Library (ab Mai) und einen **I**ntegrierten **D**ebugger **E**ditor **A**ssembler **L**inker (kurz **IDEAL**), mit dem Sie Maschinenprogramme erstellen und austesten können (ST Magazin 19, Seite 74: „Debugger der Superklasse")

Alle links abgebildeten Programme	**je DM 79,–**
IDEAL	**DM 99,–**

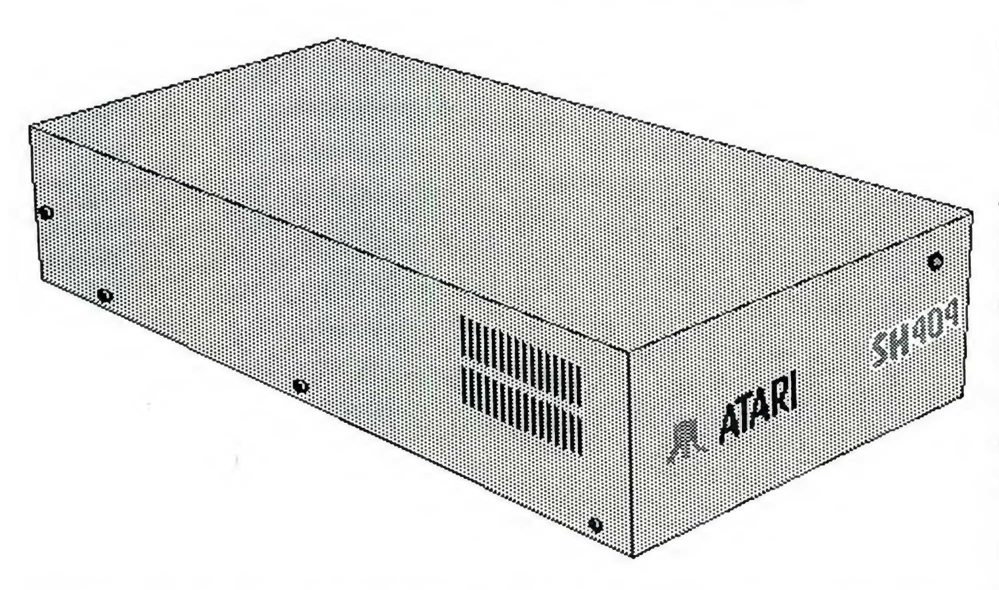
Gerne schicken wir Ihnen nähere Informationen zu – Anruf genügt!



Programme für Programmierer

OMIKRON.SOFTWARE
Erlachstraße 15 · D-7534 Birkenfeld · ☎ (0 70 82) 53 86
Frankreich: OMIKRON · France, 11 Rue Dérodé, F-51100 Reims
Luxemburg: ELECOMP, 11 Avenue de la gare, L-4131 Esch-Alzette
Niederlande: Terminal Software Publicaties, Postbus 111, NL-5110 Baarle-Nassau
Österreich: Ueberreuter Media, Laudongasse 29, A-1082 Wien
Schweiz: Thali AG, Industriestrasse 6, CH-6285 Hitzkirch

DIE FESTPLATTE TEIL 4



"Zücken Sie den Schraubendreher und werfen Sie Ihren Lötkolben schon mal an..." - So sollte eigentlich diese Folge der Artikelserie um die Festplatte anfangen. Da Sie dann aber noch einen Monat auf die notwendige Information zur Software hätten warten müssen, ziehe ich diese vor, damit wir uns in der nächsten (und letzten) Folge ausschließlich um den Umbau kümmern können.

In der letzten Ausgabe haben Sie die Verbindung zwischen dem ATARI-ST und der Festplatte in Form des Hostadapters kennengelernt. Damit allein läßt sich die SH-204 oder SH-205 aber nicht betreiben, da das Betriebssystem keine softwaremäßige Unterstützung bereitstellt. Hierzu wird bei den Festplatten eine Bootdiskette mitgeliefert, deren Inhalt wir uns jetzt näher ansehen wollen. Bei der SH-204 waren folgende Programme auf der Diskette:

```
AUTO            Ordner
HDX      .PRG   38036   Bytes
HDX      .RSC    4576     "
SHIP     .PRG    2292     "
WINCAP           1408     "
```

Bei der SH-205 sind noch einige dazugekommen, der Vollständigkeit halber liste ich sie noch einmal alle auf:

```
AUTO             Ordner
204BOOT  .IMG      512   Bytes
204ROOT  .IMG      512     "
DIRSLEFT.PRG       357     "
DIRSLEFT.TOS       416     "
FOLDR100.PRG       820     "
HDX      .PRG    38036     "
HDX      .RSC     4576     "
HINSTALL.PRG     19847     "
HINSTALL.RSC       912     "
SHIP     .PRG      206     "
WINCAP            1492     "
```

Im AUTO-Ordner selbst sollte sich das AHDI.PRG befinden, das in der letzten Version eine Größe von 3806 Bytes hat. Gegenüber der Programmversion, die bei der SH-204 mitgeliefert wurde, hat sich ein Byte geändert, was aber nicht durch eine andere Programmgröße erkennbar ist. Dadurch kommt es zu einer falschen Abfrage nach den zulässigen TOS-Versionen - normalerweise bereitet das aber keine Probleme.

Keine Hardware ohne Software

Das AHDI.PRG ist Teil des Interfaces zwischen dem ST und dem Festplatten-Controller. Die Hardware des Hostadapters habe ich in der letzten Folge erklärt. Die Software übernimmt die Kontrolle des DMA-Chips für die Arbeit mit der Festplatte und beinhaltet noch zwei Routinen für die Formatierung. Diese Teile werden aber im normalen Betrieb nicht benutzt, entweder kommen sie speziell beim Formatkommando an die Festplatte zum Zuge, oder sie sind einfach Überbleibsel aus der Urzeit, als noch mit anderen Controllern experimentiert wurde, und man das Laufwerk (durch Formatierung) erst zum Leben erwecken mußte. Dafür spricht, daß in den ersten Listings nur der Format- und der Test-Unit-Ready-Befehl auftauchen...

Wenn das AHDI.PRG (nachfolgend der Einfachheit halber Treiber genannt) startet, fragt es als erstes an allen acht möglichen Geräten am ACSI-Port nach, ob es sich dabei vielleicht um eine Festplatte handelt. Ist dies nicht der Fall, dann installiert sich der Treiber nicht und bricht den weiteren Programmverlauf ab. Wenn eine Festplatte durch einen gemeldeten Root-Sektor erkennbar ist, werden der Treiber fest installiert und die zuständigen Systemvektoren, mit denen in Programmen dann auf solche externen Geräte zugegriffen wird, auf die jeweiligen Einstiegspunkte in den Treiber umgelenkt. Bei einem Aufruf über diese Vektoren klärt der Treiber jetzt seine Zuständigkeit ab, um im Falle der Nichtzuständigkeit die Anfrage abzuweisen und den Aufrufenden freundlich

auf dem normalen Dienstweg weiterzuschicken. Im anderen Fall setzt er je nach Art der Anfrage die zuständigen Register des DMA-Chip und steuert den weiteren Datenaustausch zwischen ST und Festplatte.

HDX

Da die Festplatte vom Betriebssystem im Gegensatz zum Floppylaufwerk recht stiefmütterlich behandelt wird, müssen für die notwendigen Voreinstellungen separate Programme herhalten. Das HDX.PRG mit dem zugehörigen Resource-File und dem WINCAP-File ist dabei mit das wichtigste.
Wir haben in einer der ersten Folgen meines Kurses bereits gesehen, daß die Festplatte ebenso wie eine Diskette vor Inbetriebnahme formatiert werden muß. Neben dieser Aufgabe erledigt das HDX-Programm mehr schlecht als recht noch weitere Dinge. Durch die Seelenverwandtschaft zu MS-DOS kann - nein, könnte das Betriebssystem des STs nur Partitions bis 32 MByte verwalten. Durch einen Fehler im Betriebssystem sind bei den alten ST aber nur Partition-Größen bis 16 MByte erlaubt; in den Mega-ST-Computern soll der Fehler aber behoben sein?!

It´s my part

Eine Partition ist ein definierter Abschnitt auf der Platte, der vom Betriebssystem als eigenständiges Laufwerk interpretiert wird - wieder vergleichbar mit einer Schallplatte, auf der z.B. eine Oper komplett enthalten ist, die einzelnen Akte durch die optische Trennung aber auffindbar sind und separat angespielt werden können.
Diese Aufteilung in Partitions übernimmt ein Menüpunkt im HDX-Programm. Dabei werden über das WINCAP-File vordefinierte Aufteilungen bereitgestellt. Dieses WINCAP ist so interessant, daß ich mich in einem separaten Artikel - dort aber ausführlich - damit beschäftige (wenn es in der Setzerei keine Probleme gegeben hat, müßte er eigentlich im Anschluß an diesen Artikel zu finden sein).
Sehen wir uns jetzt noch die beiden restlichen Menüpunkte an. Mit ZERO können Sie jede Partition für sich logisch löschen. Das ist deshalb interessant, weil Sie sonst nur die Möglichkeit hätten, die Platte neu zu formatieren und zu partitionieren. Allerdings habe ich persönlich diesen Programmpunkt noch kein einziges Mal benutzt...

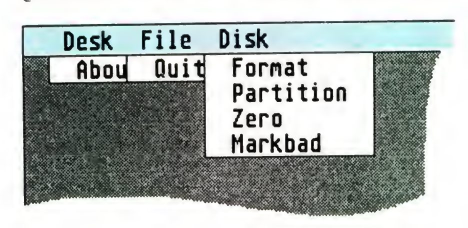


Bild 2: Die Menüleiste des HDX-Programms.

Fehlerhafte Fehlerbehandlung

Der letzte Menüpunkt wäre einer der wichtigsten - wäre, wenn er funktionieren würde. Ich glaube, selbst bei einem Laufwerk, aus dem man die Platten entfernt hat, würde dieser Programmteil 0 Fehler melden. Er sollte normalerweise das Laufwerk probelesen - was er auch macht - und dann bei einem auftretenden Read-Error diesen Sektor als defekt markieren und in der FAT als belegt kennzeichnen. Die Summe der Fehler sollte zusätzlich in einem Eintrag in der Spur 0 vermerkt werden, damit dem Laufwerk bei der Initialisierung diese Informationen zur Verfügung stehen. Denkste! Ich habe auf meiner Festplatte echte fehlerhafte Stellen, an denen der Kopfschlitten - natürlich nicht mechanisch - hängen bleibt; in meiner Defektliste (also die vom Laufwerk...) aber lautet der Fehlereintrag: 0 Fehler.

Es geht auch anders...

Fast jeder Controller stellt eine viel interessantere Fehlerbehandlung bereit. Dazu muß nach dem ersten Formatieren die Platte auf Fehler untersucht werden. Auf dem Deckel des Laufwerkes sind bereits diejenigen Defekte eingetragen, die bei der Herstellung der Platte gefunden wurden. Freuen Sie sich, wenn dort "None" steht, dann ist Ihre Platte fehlerfrei. Ansonsten sind pro MByte Kapazität ein Fehler zulässig, bei der Bruttokapazität der TM-262 von 25 MByte wären das max. 25 fehlerhafte Stellen.
Die so definierten Fehler werden nach einem bestimmten Muster in eine Defekt-Liste eingetragen. Diese darf bis zu 1024 Bytes lang sein. Mit dieser Defekt-Liste wird ein zweites Mal formatiert. Dabei versucht der Controller, die angegebenen Fehlerstellen zu umgehen, indem er innerhalb einer Spur die Information verrückt. Reicht der Platz hierzu nicht aus, setzt der Controller alle nachfolgenden Spuren um eine logische Adresse weiter und gibt dieser Stelle jetzt die Nummer, welche vorher eigentlich den Defekt adressierte. Damit bleibt die logische Reihenfolge der Blöcke erhalten; die defekte Stelle auf der Platte ist einfach nicht mehr vorhanden. Dadurch verschiebt sich natürlich auch das logische Ende der Festplatte. Das wiederum ist durch einen Sicherheitsbereich möglich, da die Angabe über die maximale Zylinderzahl vom Hersteller sehr vorsichtig berechnet wird (kommen Ihnen da nicht schon Ideen zur Vergrößerung der Festplattenkapazität?!).

Wenn einer eine Reise tut

Wie aufwendig Festplattenlaufwerke aufgebaut sind, haben Sie in der ersten Folge bereits erfahren. Wenn Sie sich dabei noch an die Problematik um den Kopfschlitten und dessen Bändigung erinnern - nun, mit dem SHIP.PRG wird die Sicherstellung des Kopfschlittens vor dem Abschalten erreicht. Dazu muß aber unbedingt bemerkt werden, daß hiermit nur der Schlitten in eine Landezone außerhalb des Datenbereiches gefahren wird! Die Köpfe landen also nach wie vor auf der Platte selbst. Ein sogenannter Headlifter findet sich in der Regel nur bei teureren Laufwerken. Die Landezone bei dem TM-262 ist lt. Handbuch auf dem Zylinder #663 - zum normalen Formatbereich bis

Zylinder #612 besteht also ein übergroßer Sicherheitsabstand. Wenn Sie sich noch einmal die beiden o.g. Inhalte der Boot-Disketten betrachten, wird Ihnen sicherlich die unterschiedliche Größe der SHIP-Programme auffallen - beide bewirken aber denselben Effekt! Die alte Routine war wohl noch in C geschrieben und enthält wiederum viele (unnütze) Teile aus der Entwicklungszeit. Das neue SHIP.PRG mit 206 Bytes ist ein reines Assemblerprogramm und erledigt seine Aufgabe ebenso akkurat. Diese Programme sollten normalerweise vor jedem Ausschalten und müssen vor jeder Bewegung (z.B. Transport) benutzt werden - es wäre doch zu schade, wenn Sie Ihre Festplatte nach dem Umzug nur deshalb wegschmeißen müßten, weil die Köpfe im Ausschaltmoment zufällig über Spur 0 standen und dort dann mal eben ein paar Bits abgekratzt haben...

DIRSLEFT - Allright

Die Programme DIRSLEFT.TOS, DIRSLEFT.PRG und FOLDERXXX. PRG sind Hilfsprogramme, die das bekannte 40-Ordner-Problem in den Griff bekommen sollen. Doch worum geht es bei diesem Problem überhaupt? Der ATARI-ST legt sich bei einem Zugriff auf einen Datenträger in Form einer Diskette oder der Festplatte eine Tabelle an, in der nach dem jeweilig ersten Aufruf die Ordner mit ihrem zugehörigen Pfad eingetragen werden. Das ist deshalb notwendig, da ja nur die ersten Ordner im Original-Directory - also in Spur 0 der Diskette - stehen. Alle weiteren Verzweigungen befinden sich wie die anderen Daten irgendwo auf dem Datenträger. Würde der ATARI-ST jetzt ohne diese Tabelle arbeiten, müßte er für jeden Zugriff auf bestimmte, in verschlungenen Pfaden angelegte, Daten zuerst wieder in Spur 0 auf den ersten Ordner zugreifen, dann auf den zweiten usw. Mit dieser Tabelle spart er sich Arbeit und Zeit und kann sich jederzeit über die einzelnen Pfade informieren. Bei der Entwicklung ist man wohl von der Überlegung ausgegangen, daß auf einer Diskette kaum mehr als 40 Pfade notwendig wären. Deswegen wird diese Tabelle auch erst gelöscht, wenn ein Diskettenwechsel erfolgt ist, vorher oder hinterher hätte das ja auch wenig Sinn - nur, wie bitteschön wollen Sie bei der ATARI-Festplatte einen Mediawechsel bewerkstelligen? Hier haben die Herren Programmierer wohl nicht zu Ende gedacht.

Aber wozu sind Programmierer sonst Programmierer, wenn sie sich nicht hinsetzten und flugs ein kleines Prögrämmchen zimmerten, das das Problem an sich zwar nicht beseitigt aber doch recht geschickt vertuscht. Mit dem FOLDRXXX.PRG wird einfach ein gewisser - nicht gerade kleiner - Speicherbereich für eine größere Tabelle reserviert. Das XXX im Programmnamen können Sie selbst durch die von Ihnen gewünschte zusätzliche Anzahl von Ordnerpfaden ersetzen; wenn Sie also der Meinung sind, daß Sie mindestens 400 Ordnerpfade bei einer Sitzung am ST benötigen - bittesehr, mit der Umbenennung in FOLDR400.PRG ist das kein Problem. Aber wie üblich kostet Sie das eine Kleinigkeit, nämlich 52800 Bytes Ihres kostbaren Speichers! Das Programm kann auch nachträglich aufgerufen werden, es muß aber im AUTO-Ordner stehen.

Wenn Sie sich nicht sicher sind, ob Sie das Programm benötigen, sehen Sie sich doch mit DIRSLEFT die verbleibende Anzahl der Ordner an. Probieren Sie ruhig ein bißchen damit herum, Sie erkennen dann selbst sehr schnell, wieviel Pfade Sie zusätzlich gebrauchen können. Um es noch einmal klarzustellen: Ordner auf der Festplatte können Sie soviel wie nötig anlegen (natürlich durch die Kapazität begrenzt) - nur wenn Sie in alle während einer Sitzung mal hineinschauen wollen...

Sollten Sie das FOLDRXXX.PRG noch nicht besitzen - dafür aber der stolze Besitzer eines Akkustikkopplers sein - so können Sie dieses Programm und auch die neuen Treiber über die ATARI-Mailbox (Tel. 06142/21161) abrufen.

Buhten

Da der oben beschriebene Treiber für die Festplatte nicht im Betriebssystem Ihres STs integriert ist, muß er beim Einschalten jedesmal neu aufgerufen werden. Die normale und langsame Arbeitsweise ist das Benutzen der Boot-Diskette, in deren AUTO-Ordner sich das AHDI.PRG befinden muß. Aber warum soll - was bei der Diskette ja funktioniert - nicht auch die Platte autobootfähig sein? Sieht man sich einmal die Initialisierungsroutinen des Betriebssystem an, so findet man dort tatsächlich eine Abfrage nach einer bootbaren Festplatte. Nur - um mit der Festplatte arbeiten zu können, muß doch der Treiber installiert sein, oder?!

Die Festplatte besitzt in Spur Null - ebenso wie die Diskette - einen Bootsektor, aus dem ein Programm geladen und gestartet werden kann. Außerdem befindet sich hier noch die Information über die Aufteilung der Platte in Partitions, die Größe der Partitions mit ihren zugehörigen Startclustern, einen Vermerk über die Autoboot-Möglichkeit einer Partition (!) und die oben erwähnte, derzeit sinnlose Fehlerliste. Vor diesen ganzen Daten ist Platz für ein Prögrämmchen von 448 Bytes. Findet der ATARI bei der Initialisierung hier am Anfang das berühmte magische Wort, dann verzweigt er die Startinitialisierung hierher und versucht das dort stehende Programm zu starten. Leider ist unser Treiber mit seinen 3806 Bytes aber etwas größer - was also tun?

In der Liste der Programme, die auf der Boot-Diskette für die SH-205 zu finden sind, sehen Sie zwei Files mit den Bezeichnungen 204BOOT.IMG und 204ROOT. IMG. Dabei handelt es sich um zwei Minitreiber, die durch das Programm HINSTALL auf der Festplatte in Block 0 (204ROOT.IMG) und Block 1 (204BOOT.IMG) installiert werden. Block 1 ist der erste Block der Partition 0, deswegen kann mit diesem Treiber die Festplatte auch nur für die Partition C autobootfähig gemacht werden. Jetzt installiert HINSTALL noch den Treiber als Datei mit dem

Namen SH204DVR. SYS in Partition C, diese wird beim Booten dann durch die zwei Minitreiber aufgerufen, und damit wird der vollständige Treiber installiert. So einfach geht das. Berechtigterweise können Sie jetzt einwerfen, daß die beiden IMG-Files ja jeweils 512 Bytes lang sind und somit nach meiner Aussage nicht in die 448 Bytes passen - stimmt, anscheinend liest HINSTALL erst die Information zur Festplatte aus Block 0 aus, schreibt anschließend das IMG-File und überschreibt dieses File am Ende wieder mit den Daten der Festplatte. Das funktioniert, da sich am Ende der beiden Files nur eine - für den Programmablauf unnötige - Symboldatei befindet.

Zum Schluß...

möchte ich noch einmal auf den nachfolgenden Artikel zum WINCAP-File verweisen. Nur durch dieses File sind wir in der Lage, Modifikationen an unserer Festplatte ohne großartige programmiertechnische Klimmzüge durchzuführen. In der nächsten Folge - ich verspreche es hoch und heilig - werden wir dann wirklich zum Schraubendreher greifen. Werden Sie sich aber jetzt schon darüber klar, daß Sie damit jegliche Garantieansprüche über den Haufen werfen. Sollten Sie Ihre Festplatte erst vor kurzem gekauft haben, schauen Sie lieber noch einmal auf dem Rechnungsbeleg nach, wann sie wahrscheinlich sowieso kaputt geht - lt. einem Gesetz von Murphy ist das genau ein Tag nach Ablauf der Garantiezeit...

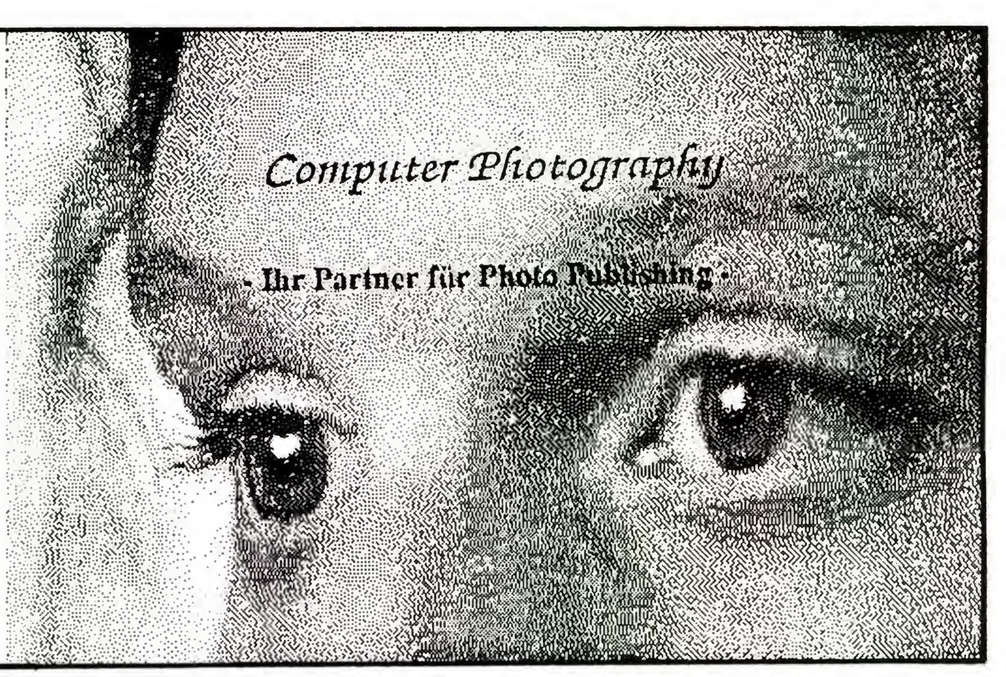
hp

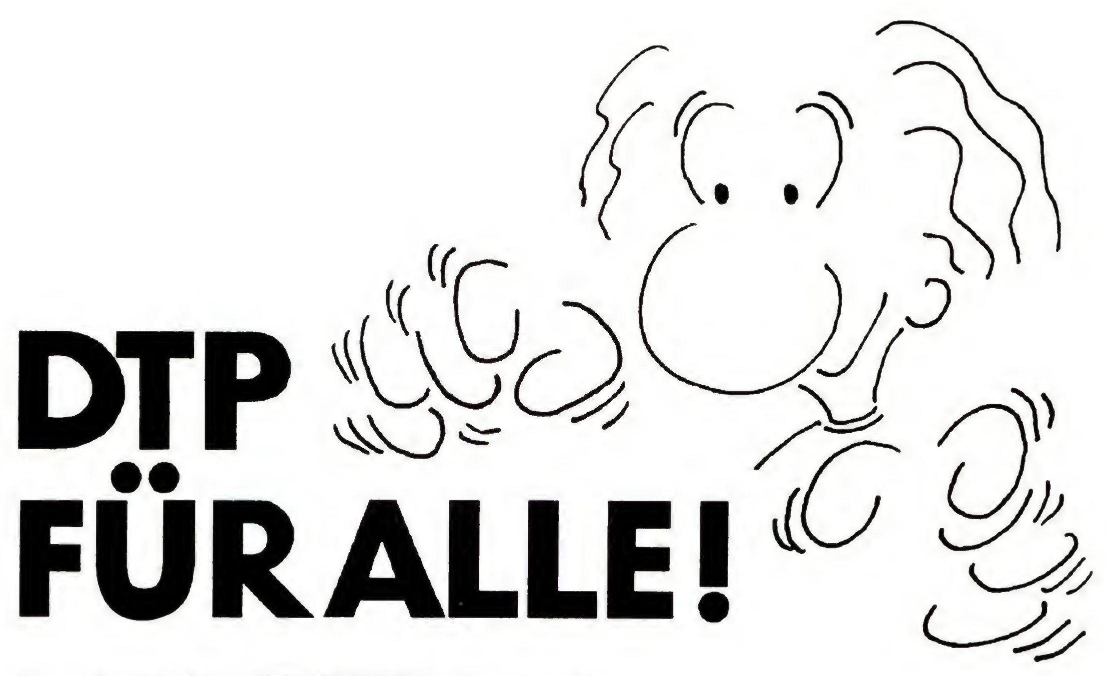
In des Autors eigener Sache...

Ich habe auf der CeBIT einiges an Kritik zu meiner Grundlagenserie "Die Festplatte" gehört und möchte deshalb ein paar Erklärungen geben, da sicher auch andere Leser den gleichen Zorn haben. Hauptkritikpunkt war die ewige Hinausschieberei der Umbauanleitung. Stimmt, nur so umfangreich wie die Serie geworden ist, war sie von mir am Anfang nicht geplant (Es ist mein allererstes Werk als Schreiberling). Das hat sich erst im Laufe der Recherche so ergeben. Außerdem mußte ich jetzt noch einmal schieben, um die Software vorzuziehen - was nutzt Ihnen ein umgebautes Laufwerk, wenn die Information zur Formatierung einen Monat später kommt?! Dazu kommt noch, daß aufgrund des Platzmangels in Heft 4 mein Artikel rausgeflogen ist...(kein Kommentar).
Also - Teil 4 haben Sie jetzt vorliegen und im nächsten Heft kommt definitiv der Umbau. Apropos Umbau - da scheint irgendwo noch ein Mißverständnis zu kursieren - es wird kein zweites Laufwerk eingebaut, sondern nur das alte gegen ein neues Laufwerk mit höherer Kapazität ausgetauscht.
Wenn Sie noch am Zweifeln sind, welches Laufwerk Sie benutzen können, warten Sie den alleraller-letzten Teil 6 ab! Und - im Falle von Fragen, Anregungen oder Kritik - schreiben Sie bitte an die Redaktion; damit solch Zorn nicht Ihre Energien verbraucht...

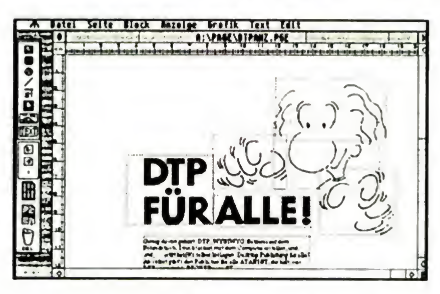
hp

ENDE





Genug davon gehört: DTP, WYSIWYG, Setzerei auf dem Schreibtisch, Drucksachen mit dem Computer erstellen, und, und, ... Jetzt heißt's selber loslegen: Desktop Publishing für alle! Ab sofort gibt's den Publisher für alle ATARI ST, der hält, was DTP verspricht: BECKERpage ST.
Sie wollen es nicht glauben? Sehen Sie selbst: diese Anzeige wurde komplett mit BECKERpage ST realisiert.
BECKERpage ST hat nicht nur alleTextverarbeitungsfunktionen drauf, schafft nicht nur verschiedene Schriftarten wie Helvetica, Times und Courier in allen Variationen (fett, kursiv, negativ, outline, gesperrt), kann nicht nur eigene Schriftbilder entwerfen (mit dem integrierten Schrifteditor FOMA) - nein, BECKERpage ermöglicht auch die Kreation und Integration beliebiger Grafiken.Die Illustration dieser Anzeige wurde mit dem mitgelieferten Grafikprogramm PROFI PAINTER erstellt. Ebensogut können Grafiken und Bildschirmausschnitte aus anderen Programmen übernommen und beliebig variiert werden. Dafür sorgt die integrierte Snap-Shot-Accessory PAINTER FOTO (siehe Hardcopy links in Anzeige).



Nicht genug. BECKERpage ST ermöglicht vollständiges Seitenlayout am Bildschirm. Natürlich mit der Maus: mehrspaltiger Seitenumbruch, Blindtextfunktion, Picture Wrap, beliebig abgestufte Raster, Kreise, Linien, Rechtecke, und, und ...
Genug gehört? Dann ran an DTP! BECKERpage ST macht's möglich.
Minimalkonfiguration: ATARI ST mit ROM TOS und 1 MByte Speicher, ATARI-Laserdrucker oder NEC P6, P7, P 2200, Star NB 15, Epson 9- und 24-Nadeldrucker und Kompatible.

BECKERpage ST

Version 1.0,
inkl. PROFI PAINTER nur DM **398,-**

Gleich anfordern

Legen auch Sie Wert auf perfekte Gestaltung bei gleichzeitiger Kostenreduzierung? Dann fordern Sie unverbindlich unseren kostenlosen Prospekt zu BECKERpage ST an. Hier finden Sie alles Wissenswerte über dieses leistungsstarke DTP-Programm.

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

COUPON

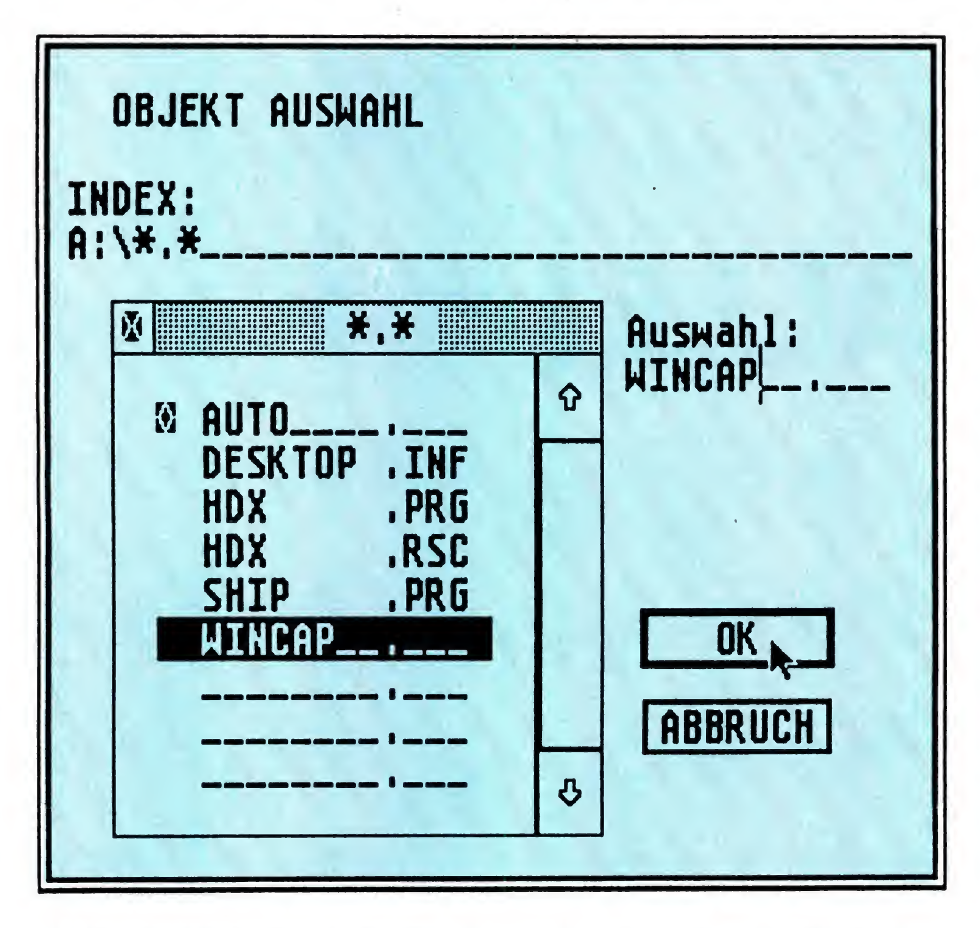
Hiermit bestelle ich

Name, Vorname

Straße

Ort

DAS WINCAP-FILE

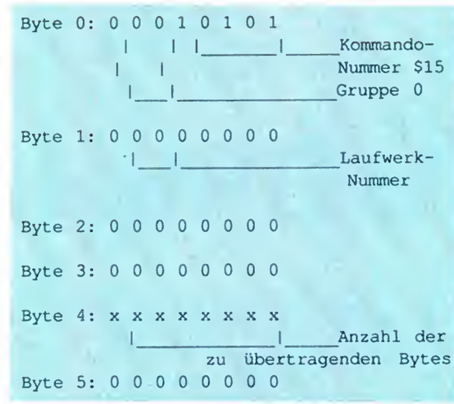


Man müßte es eigentlich als sträfliche Nachlässigkeit ansehen, daß dieses unscheinbare Anhängsel an das HDX.PRG von der bisherigen Literatur zur Festplatte so stiefmütterlich behandelt wurde. Mit dieser kleinen Bit-Sammlung kann man nämlich herrlich in der Festplatte herumspielen...

Die Festplatte muß - wie wir bereits erfahren haben - vor der ersten Benutzung eingestellt, d.h. formatiert werden. Da es natürlich unterschiedliche Größen - also Formate - gibt, müssen wir diese Werte irgendwie an den Controller übermitteln. Nur mit diesen Daten kann er dann z.B. die Ansteuerung der einzelnen Köpfe vornehmen. Ich habe in der dritten Folge der Serie um die Festplatte die Kommandos des Controllers aufgeführt. Darunter befindet sich der Befehl MODE SELECT ($15). Damit kann eine Parameter-Datei übertragen werden, anhand derer der Controller die Voreinstellung der Festplatte durch eine anschließende Formatierung vornimmt. Diesen Befehl sehen wir uns zunächst einmal näher an.

$15

Das MODE SELECT COMMAND sieht wie folgt aus:



```
Byte 0: 0 0 0 1 0 1 0 1
        |   | |_______|_____Kommando-
        |   |               Nummer $15
        |___|_______________Gruppe 0

Byte 1: 0 0 0 0 0 0 0 0
         |___|______________Laufwerk-
                            Nummer

Byte 2: 0 0 0 0 0 0 0 0

Byte 3: 0 0 0 0 0 0 0 0

Byte 4: x x x x x x x x
        |_____________|_____Anzahl der
              zu übertragenden Bytes
Byte 5: 0 0 0 0 0 0 0 0
```

In Byte 4 wird die Anzahl der im Anschluß an die Command-Phase übertragenen Bytes bekanntgegeben. Es müssen mindestens 12 Bytes sein; es können aber auch bis zu 22 Bytes bei softsektorierten und 24 Bytes bei hardsektorierten Laufwerken werden. Diese ersten 12 Bytes beinhalten die MODE-SELECT-PARAMETER-LIST und die EXTENT-DESCRIPTOR-LIST.

Die Parameter-Liste umfaßt vier Bytes mit den Werten 0, 0, 0 und 8 und gibt nur Auskunft über die Länge der nachfolgenden EXTENT-DESCRIPTOR-LIST. Das hört sich natürlich total unsinnig an, aber damit sollte eine spätere Erweiterung möglich sein.

Die acht Bytes der EXTENT-DESCRIPTOR-LIST enthalten Informationen über die Datendichte, die Anordnung, daß das ausgewählte Laufwerk formatiert werden soll und die Angabe über die Größe der Datenblöcke (der Wert muß zwischen 256 und 1024 liegen). Die beiden erstgenannten Information werden übrigens durch den Wert 0 ausgedrückt. Die eigentliche Liste mit den Formatierungsdaten wollen wir uns wieder etwas genauer ansehen.

Byte 0 - *List Format Code*
Hier muß entweder 0 für softsektorierte oder 1 für hardsektorierte Laufwerke stehen.

Byte 1 *(MSB - Most significant, also höherwertiges Byte) und*
Byte 2 *(LSB - Low significant Byte) - Cylinder Count*
Die beiden Bytes beschreiben die Anzahl der Zylinder auf der zu formatierenden Festplatte.

Byte 3 - *Data Head Count*
Die Anzahl der Köpfe.

Byte 4 *(MSB) und*
Byte 5 *(LSB) - Reduced Write Current Cylinder*
Da der Kreisumfang der einzelnen Zylinder zur Mitte der Platte hin abnimmt, rücken die magnetischen Informationen auf der Platte entsprechend enger zusammen. Damit das

geht weiter...

magnetische Feld kein benachbartes Bit beeinflußt, wird der Schreibstrom verringert. In diesen beiden Bytes wird jetzt die Nummer des Zylinders eingetragen, ab der diese Stromverminderung einsetzt.

Byte 6 *(MSB) und*
Byte 7 *(LSB) - Write Precompensation Cylinder*
Hier wird nun eine weitere Datensicherung eingebaut, die ich in Teil 3 der Festplattenserie schon kurz beschrieben habe. Die Vorkompensation verrückt die Flanken der Takt- und Datenfelder so, daß diese Informationen beim Lesen richtig interpretiert werden. Es wird aber erst auf den inneren Spuren akut, daher bestimmen diese beiden Bytes die Zylindernummer für den Einsatz der Vorkompensation.

Byte 8 - *Landing Zone Position*
Mit diesem Byte wird die Landespur definiert, auf die der Kopfschlitten durch den START-/STOP-Befehl gefahren wird. Mit einer "1" in Bit 7 parkt der Schlitten außerhalb von Spur 0, mit einer "0" fährt er die Köpfe in die Mitte der Platte. Die genaue Landeposition errechnet sich durch die Addition der Zylinderanzahl und dem Wert, der in Bit 0 bis 6 steht.

Byte 9 - *Step Pulse Output Rate Code*
Wie beim Diskettenlaufwerk kann für die Schrittgeschwindigkeit des Kopfmotors eine unterschiedliche Steprate eingestellt werden. Das betrifft allerdings nur den SEEK-Befehl. Die drei verschiedenen Werte hierfür sind:
0 = 3,0 ms (ungepuffert)
1 = 28 µs (gepuffert)
2 = 12 µs (gepuffert)

Die Bytes 10 und 11 sind hier nicht so wichtig, da sie nur Angaben für hardsektorierte Festplatten beinhalten.

Klartext

Für den normalen Benutzer der Festplatte wäre es natürlich viel zu aufwendig, jedesmal den doch recht komplizierten Ablauf der Programmierung vorzunehmen. Deshalb haben die Entwickler der Festplatte dem Benutzer ein Programm zur Verfügung gestellt, mit welchem er diese Aufgabe einfach hinter sich bringt. Dieses Programm heißt HDX.PRG und wird auf der Boot-Diskette mitgeliefert. Vorweg möchte ich sagen, das innerhalb des Programmes bereits die Standardwerte, wie wir sie nachher noch sehen werden, eingetragen sind. Nichtsdestotrotz fragt das HDX-Programm nach dem Start das Vorhandensein des WINCAP-Files ab. Fragen Sie mich bitte nicht, was WINCAP bedeuten soll, ich weiß es nicht. Viel interessanter ist ja auch, was man mit dem Daten-File für das HDX-Programm alles anstellen kann. Doch betrachten wir uns zunächst einmal das Original.
Im nebenstehenden Kasten finden Sie das gleiche WINCAP, wie es z.Zt. mit der SH-205 ausgeliefert wird. Grundsätzlich ist dazu zu sagen, das alle Kommentarzeilen an der ersten Stelle einer Zeile ein #-Symbol haben. Die Parameter dagegen werden von zwei Doppelpunkten eingeschlossen. Theoretisch können wir also ein kurzes File erstellen, dessen Daten nur aus Informationen zwischen den Doppelpunkten bestehen. Gehen wir jetzt aber das File Schritt für Schritt durch.

```
# Hard disk format and partition configuration file.
# version 1.1: 29-May-1986 lmd
# version 1.2: 23-Jun-1987 lmd          SH205 menu entry

#
# Global parameters:
#       ms#count         Maximum partition size in sectors
#                        (***DON'T CHANGE***)
#
:@@=Parameters:ms#32767:

#
# Format parameters:
# 'mn' entries (default values in [brackets]):
#       mn=name          appears in menu, name of format parms
#       hd#count         number of heads [4]
#       cy#count         number of cylinders [612]
#       lz#n             landing zone [4]
#       rt#n             seek rate code [2]
#       in#n             interleave factor [1]
#       rw#cylno         reduced-write current cylinder [cy + 1]
#       wp#cylno         write precomp cylinder [cy + 1]
#       pt=name          name of default partition to use [4-6-10]
#       sp#count         number of sectors/track [17]
#
20 Mb Mitsubuishi        :mn=SH204:
10 Mb Mitsubuishi        :mn=SH104:hd#2:pt=10:
SH204 in plastic case    :mn=SH205:

#
# Partition selections
# 'pr' entries ('N' is '0', '1', '2', or '3'):
#       pr=name          name of partition
#       pN#size          size of partition N in bytes [0]
#       fN#number        flag byte for partition N [0 if no pN,
#                        1 otherwise]
#       iN=ccc           id bytes for partition N [GEM]
#
# Numbers are in bytes; may be followed by 'k' (x1024) or
#                        'm' (megabytes).
#
:pr=4-6-10:p0#4424k:p1#6m:p2#10m:
:pr=10-10:p0#10404k:p1#10404k:
:pr=16-4:p0#16383k:p1#4425k:
:pr=4-16:p1#16383k:p0#4425k:
:pr=5-5-10:p2#10568k:p0#5m:p1#5m:
:pr=5-10-5:p0#5m:p1#10568k:p2#5m:
:pr=10-5-5:p0#10568k:p1#5m:p2#5m:
:pr=5-5-5-5:p0#5448k:p1#5m:p2#5m:p3#5m:
:pr=10:p0#10m:
:pr=5-5:p0#5m:p1#5m:
```

Tabelle 1: So wird das WINCAP-File mit der SH205 ausgeliefert.

Hard disk format and partition configuration file

So die Überschrift und wohl offizielle Bezeichnung dieses Files.

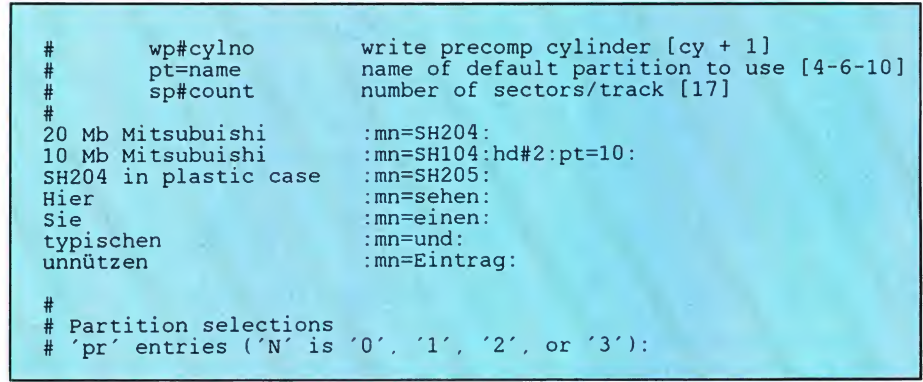
Version

Hier werden die unterschiedlichen Versionen des Files eingetragen. Für eine fortlaufende Veränderung ist ein solcher Eintrag eigentlich unumgänglich. Wenn Sie dieses File selbst verschiedentlich modifiziert haben, sind Sie irgendwann bestimmt froh, die einzelnen Versionen durch solche Kommentarzeilen noch auseinanderhalten zu können. Machen Sie davon also regen Gebrauch!

Global Parameters

Es gibt gewisse Standardwerte, die sich normalerweise nicht ändern. Ein Beispiel dafür ist der Eintrag ms#count. Mit diesem Wert wird die maximale Größe einer Partition als Anzahl der Sektoren übergeben. Der Eintrag wurde notwendig, da bei dem älteren Betriebssystem ein Fehler die Einrichtung von Partitions bis 32 MByte verhindert. Bei den Mega-STs soll dieser Fehler behoben sein, trotzdem steht hier noch der Wert ms#32767. Wenn Sie jetzt nachrechnen wollen:

32.767 Sektoren x 512 Bytes = 16.776.704 Bytes



```
#       wp#cylno         write precomp cylinder [cy + 1]
#       pt=name          name of default partition to use [4-6-10]
#       sp#count         number of sectors/track [17]
#
20 Mb Mitsubuishi        :mn=SH204:
10 Mb Mitsubuishi        :mn=SH104:hd#2:pt=10:
SH204 in plastic case    :mn=SH205:
Hier                     :mn=sehen:
Sie                      :mn=einen:
typischen                :mn=und:
unnützen                 :mn=Eintrag:

#
# Partition selections
# 'pr' entries ('N' is '0', '1', '2', or '3'):
```

Tabelle 2: So wird's gemacht!

d.h. unsere Partitions können eine Größe bis 16 MByte annehmen. Wozu die vorherige Eintragung von ":§§=Parameters:" dient, weiß ich nicht - da muß ich leider passen.

Format parameters

Dafür können wir uns aber jetzt so richtig austoben. Die folgende Liste enthält nämlich genau die Parameter, welche mit dem MODE-SELECT-Befehl an den Controller übertragen werden. Die nachstehenden Daten liegen bereits als - sagen wir einmal Mindesteinsatz - in dem HDX.Prg in den Data-Zeilen am Schluß des Programms vor. Das HDX wäre also theoretisch nicht auf das WINCAP angewiesen. Doch sehen wir uns diese Tabelle wieder in der Reihenfolge an.

mn=name

Die Programmierer des HDX.PRG haben sich hier etwas feines ausgedacht. Normalerweise wird in einem GEM-Programm die Information über die Bildschirmmaske im Resource-File abgelegt. Das dient u.a. dazu, daß man Programmänderungen oder Ausführungen in anderer Sprache nur durch den Neueintrag im Resourcefile erreicht. Nichts anderes geschieht soweit im HDX-Programm. Doch hier ist man einen Schritt weitergegangen.
Im HDX.RSC wird zwar der Maskenaufbau bereitgestellt, doch die Texte für die Auswahl der verschiedenen Versionen werden nicht hier sondern im WINCAP eingetragen. Dadurch kann man jetzt auch so herrliche und unnütze Spielchen, wie sie im nebenstehenden Bild zu sehen sind, treiben. Sehen Sie sich im Ausschnitt des WINCAP die Modifizierung an und vergleichen Sie damit das Resultat im Programmablauf. Sie können hier bis zu 16 verschiedene Einträge vornehmen. Ich habe mir damit ein WINCAP geschrieben, in dem an dieser Stelle die verschiedenen Laufwerkstypen mit unterschiedlichen Formatangaben eingetragen sind. Das Beispiel hat aber noch keine Auswirkung auf die Formate selbst, dazu müssen noch weitere Parameter eingetragen werden.



Bild 2: Das Format-Menü läßt sich leicht mit WINCAP manipulieren

hd#count

Hiermit wird die Anzahl der Köpfe unserer Festplatte eingestellt. Der Default-Wert, d.h. der voreingestellte Wert ist 4.

cy#count

Hiermit wird die Anzahl der Zylinder angegeben. Die Werte brauchen dazu nicht in hexadezimale Angaben umgerechnet zu werden, das erledigt das Programm. Der Default-Wert ist

nicht vergessen wird, führt das HDX-Programm schon mal eine Einteilung durch, diese kann ja jederzeit wieder geändert werden. Der Name für diese Partitionsaufteilung kann mit pt=name bestimmt werden. Wichtig ist nur, daß für diesen Namen auch in der weiter unten stehenden Liste die Daten eingetragen sind.

sp#count

Hiermit wird die Anzahl der Sektoren pro Spur bestimmt. Der Default-Wert ist 17. Wie Sie bei dem Eintrag des Interleave-Faktors gesehen haben, könnte man hier auch 18 Sektoren verwenden, dann allerdings nur mit einem Interleave >1 (z.B. 2).

Partitionierung

Nachdem die Default-Werte im WINCAP beschrieben sind, finden wir die ersten Variationen. Auch wenn es verwunderlich erscheint, was hier ein 10-MByte-Laufwerk zu suchen hat - ATARI hatte ganz zu Anfang wirklich einmal vor, eine solche Spardose zu bringen. Über die Entwicklungszeit sind dann die Preise aber so stark gefallen (es hat halt doch eine ganze Weile gedauert...), daß zwar dieser Laufwerkstyp, aber nicht der Eintrag unter den Tisch gefallen ist. Uns bietet er einen Hinweis, wie wir andere Daten eintragen können:

:mn=SH104:hd#2:pt=10:

```
#       iN=ccc          id bytes for partition N [GEM]
#
# Numbers are in bytes; may be followed by 'k' (x1024) or
#                       'm' (megabytes).
#
:pr=4-6-10:p0#4424k:p1#6m:p2#10m:
:pr=groß:p0#10404k:p1#10404k:
:pr=16-4:p0#16383k:p1#4425k:
:pr=KLEIN:p1#16383k:p0#4425k:
:pr=5-5-10:p2#10568k:p0#5m:p1#5m:
:pr=D I C K:p0#5m:p1#10568k:p2#5m:
:pr=dünn:p0#10568k:p1#5m:p2#5m:
:pr=5-5-5-5:p0#5448k:p1#5m:p2#5m:p3#5m:
:pr=10:p0#10m:
:pr=5-5:p0#5m:p1#5m:
```

Listing 3: Hier werden die Daten zur Positionierung eingetragen

Dazu müssen Sie noch wissen, daß das kleinere Laufwerk statt 2 nur 1 Platte besitzt, die restlichen Daten (bis auf die Kapazität) aber identisch sind. Damit ist auch sofort klar, was hier geändert wird: erstens der Name für die Auswahlbox, zweitens statt vier nur zwei Köpfe und drittens eine andere Partition.

Nach diesem Muster können Sie jetzt beliebige Laufwerke formatieren, als Beispiel nachfolgend die Daten, wie ich sie für das Laufwerk TM-755 einstelle:

:mn=SH404:hd#5:cy#981:pt=
16-16-8:

Für den letzten Eintrag muß natürlich auch eine entsprechende Formataufteilung in der Liste angegeben sein.

How much

Die so übertragenen Formatinformationen werden in Block 0 auf der Platte abgelegt. Die Eintragungen beinhalten im einzelnen:

- Anzahl der Zylinder
- Anzahl der Köpfe
- Nummer des Reduced-Write-Current-Cylinders
- Nummer des Write-Precompensation-Cylinders
- Landeposition relativ zum letzten Zylinder
- Seek-Rate
- Interleave-Faktor
- Anzahl der Sektoren pro Spur
- und die Gesamtgröße der Festplatte

Danach folgen die Informationen über die einzelnen Partitions. Der Platz reicht für vier Eintragungen. Diese sind:

- das Flagbyte: Es gibt an, ob diese Partition ein- oder ausgeschaltet ist und ob von hier gebootet werden kann.
- die ASCII-Zeichen "GEM", wenn es sich um eine Partition unter GEMDOS handelt
- die Blocknummer des ersten Sektors der Partition und
- die Länge der Partition in Blöcken

Am Ende von Block 0 finden wir noch zwei Einträge:

- die Nummer des Blockes, ab dem die Fehlerstellen aufgelistet sind und
- die Länge dieser Fehlerliste. Steht hier der Wert 0, dann ist die Platte angeblich fehlerfrei.

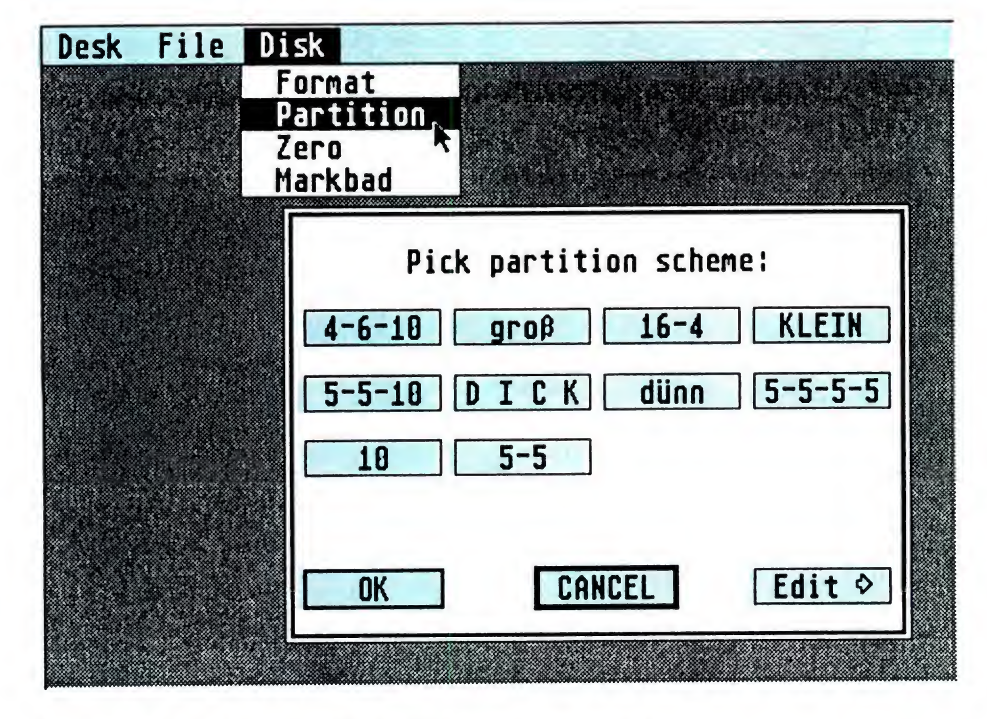


Bild 3: Auch die Partitionierung ist gegen Sabotage nicht gefeit.

612. Schlagen wir in den Handbüchern der Laufwerke nach, die bisher von ATARI verwendet wurden, so finden wir dort einen Wert von 615 Zylindern. Warum werden also die

3 Zyl. x 17 Sekt. x 4 Köpf. x
512 Byte/Sekt. = 104.448 Bytes

unterschlagen? Die Antwort können Sie im WINCAP-File nachlesen. Dort steht für die SH-204 als Laufwerk noch ein 20-MByte-Typ von Mitsubuishi - und die haben wirklich 612 Zylinder. Selbst bei der letzten Änderung vom 23.06.87 scheint das bei ATARI noch niemand aufgefallen zu sein...

lz#n

Wie oben beschrieben, wird damit der Landezylinder in Relation zum ersten oder letzten Data-Zylinder angegeben. Hier ist der Default-Wert 4, d.h. die Köpfe landen innen (Bit7=0) auf Zylinder #616 bzw. #619. Bei Tandon finden wir als Landezylinder für die TM-262 die Nummer 663, bei Seagate für die ST-225 den Wert 670. Dazwischen haben wir reichlich Platz, um die Kapazität noch ein bißchen zu erhöhen. Ähnlich wie bei den Diskettenlaufwerken geht Probieren über Studieren, hier muß jeder die Möglichkeit selbst austesten. Mir ist zwar noch nicht zu Ohren gekommen, daß dabei Probleme oder Defekte aufgetreten sind, aber das Risiko müssen Sie natürlich selber tragen. Mir sind Laufwerke bekannt, die mit 660 Zylindern formatiert wurden und einwandfrei funktionieren. Das ist dann immerhin eine Kapazität von ca. 23 MByte.

rt#n

Hiermit stellt man den Kode für die Steprate beim SEEK-Befehl ein, er ist normalerweise 2 und steht damit schon auf der schnellsten Geschwindigkeit. Mit dem SEEK-Befehl wird eine absolute Adresse auf der Platte angegeben. Während der Kopfschlitten auf diese Position fährt, kann der SCSI-Bus freigegeben werden, um anderen Teilnehmern den Datenverkehr zu ermöglichen. Das funktioniert bei den ATARI-Systemen leider nicht.

in#n

Der Interleavefaktor kann mit diesem Kürzel verändert werden. Wir verwenden hier normalerweise einen Interleave von 1. Das bedeutet, daß die komplette Spur innerhalb einer Umdrehung der Platte gelesen wird. Dadurch erreicht man die schnellstmögliche Datenübertragungsrate. Es gibt aber Systeme (z.B. IBM-PCs), deren Bus nicht so schnell arbeitet. Mit einem Interleave von >1 verlangsamt sich die Übertragung, da jetzt zwei oder entsprechend mehr Umdrehungen pro Spur benötigt werden. Für uns läßt sich hiermit - zugegebenerweise auf Kosten der Geschwindigkeit - nochmal die Kapazität erhöhen. Wie bei der Diskette muß jeder Sektor auf der Platte von einer bestimmten Anzahl sogenannter Synchronisationsbytes eingeläutet werden, damit unser Controller den Sektoranfang erkennen kann. Für einen Interleave (I) von 1 sind dazu sicherheitshalber mehr Bytes notwendig als bei einem Faktor >1. Insgesamt beträgt die Sektorlänge für I=1 576 Bytes und für I>1 566 Bytes. Da eine Spur etwa 10.500 Bytes aufnehmen kann, paßt bei I>1 ein Sektor mehr auf die Spur, so daß wir die Kapazität von

z.B. 17 x 4 Köpf. x 512 Bytes/S
x 615 Sekt.= 21.411.840 Bytes
auf
18 x 4 x 512 x 615 =
22.671.360 Bytes

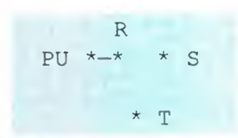
erhöhen können. Das funktioniert wirklich, probieren Sie es einmal aus.

rw#cylno

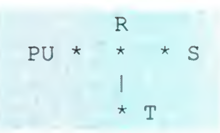
Ab diesem Zylinder wird normalerweise der Schreibstrom vermindert. In unserem WINCAP sehen Sie als Eintrag cy+1. Das würde ja bedeuten, daß unser Laufwerk in jedem Fall erst nach dem offiziellen Ende des Datenbereiches dieses Feature benutzt. Das liegt ganz einfach daran, daß die heutigen Laufwerke über ihren Mikroprozessor selbst den Schreibstrom kontrollieren. Damit ist eine Steuerung über den Controller hinfällig.

wp#cylno

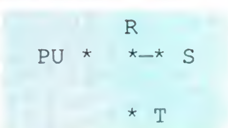
Bei der Vorkompensation steht ebenfalls als Zylindernummer für den Start cy+1. Hier wird die Steuerung nicht beeinflußt, weil auf der Controllerplatine eine kleine Steckbrücke die Einstellung vornimmt und absolute Priorität vor der softwaremäßigen Angabe hat. Diese Brücke hat die Form eines T, da sie insgesamt drei Möglichkeiten zuläßt, aber jeweils nur eine selektiert werden darf. Man kann diese Steckleiste leicht an der Buchstabenkombination erkennen. Die Einstellungen bedeuten im einzelnen:



Die Vorkompensation ist für beide Laufwerke ausgeschaltet. ATARI gibt diese Stellung für das Tandon-Laufwerk an.



Die Vorkompensation wird für beide Laufwerke auf allen Spuren eingeschaltet.



Die Vorkompensation startet ab der Zylindernummer, ab der auch der Schreibstrom reduziert wird. Das trifft für das Seagate-Laufwerk zu.

pt=name

Nach dem Formatieren muß die Festplatte noch partitioniert werden. Wird dieser Vorgang versäumt und der Computer zwischenzeitlich ausgeschaltet, bekommen Sie gewaltigen Ärger. Da die Festplatte als solche beim Booten zwar erkannt wird, der Treiber aber dann einen Fehler interpretiert - weil er keine Partition findet - führt er einen System-Reset durch. Aus dieser Schleife kommt man nur durch Klimmzüge wieder heraus (aber es geht). Damit die Partitionierung nach dem Formatieren

Die Angaben, soweit sie die Partition-Aufteilung betrifft, können wir im WINCAP-File ebenfalls modifizieren:

pr=name

Das ist nur der Name der Partition. Neben einer sinnvollen Beschreibung können hier auch wieder reine Phantasienamen benutzt werden. Das entsprechende Beispiel sehen Sie im nebenstehenden Bild.

pN#size

Hiermit wird jetzt die Größe einer Partition bestimmt. Als Ersatz für den Buchstaben "N" muß natürlich die Nummer der Parttion eingesetzt werden und zwar Werte zwischen 0 (entspricht später Laufwerk C) und 3 (=Laufwerk F). Die Größe selbst kann entweder in MByte mit dem Zusatz "m" oder in KBytes mit dem Zusatz "k" angegeben werden. Dabei ist

1m = 1024k = 1.048.576 Bytes

fN#number

Hiermit könnte man die Partition ein- oder ausschalten. Das wird vom Programm aber alleine erledigt, da es ja aus der Liste die Anzahl der Partitions entnehmen kann.

iN=ccc

Auch hier sollte man nach Möglichkeit nichts ändern, da der Default-Wert mit den Buchstaben G, E und M bereits die richtigen Daten enthält.

Spieltrieb

Nach der Erklärung finden Sie am Schluß der WINCAP-Datei bereits eine Auswahlliste für verschiedene Aufteilungen Ihrer Festplatte. Hier muß auf jeden Fall der Eintrag vorhanden sein, der über den Default-Wert von pt=name angesprochen wird. Andere Einteilungen können Sie auch in der Menüwahl des HDX.PRG über das Fenster "Menu" von Hand frei bestimmen. Da zu Ihrer Sicherheit vor der endgültigen Partitionierung noch einmal eine Sicherheitsabfrage erscheint, können Sie ruhig ein bißchen herumspielen; achten Sie dabei bloß auf Ihren rechten Zeigefinger, daß er keinen Murks baut - für Datenverluste jeglicher Art übernehme ich keine Garantie.

hp

ENDE



Wußten Sie eigentlich, wieviel Arbeit Ihnen Ihr ST bei den täglichen Büro-Aufgaben abnehmen kann? Immer vorausgesetzt, daß Sie über eine Software verfügen, die alle nötigen Funktionen besitzt, komfortabel und schnell ist; wie die von der GFE R. Becker KG entwickelten Programme der Reihe SYBEX ST-Kontor (jeweils mit Trainingsbuch). Ihr ATARI ST (260–1040, MegaST 2–4) sollte einen Speicher von 1 MB RAM (für TOS-Manager und Kundenverwaltung reichen auch 512 KB) und das Betriebssystem TOS im ROM haben. Die mächtige FiBu benötigt ein double-sided Laufwerk mit 1 MB, optimal wäre eine Festplatte.

Natürlich finden Sie bei SYBEX noch mehr Software und Bücher zu Ihrem Rechner. Fordern Sie einfach unseren ATARI ST-Prospekt oder unser Gesamtverzeichnis an.

So hilft Ihnen der ST-Kontor TOS-Manager:

Er ist die neue Verwaltungszentrale Ihres ATARI-Rechners. Von hier aus können Sie alle übrigen ST-Kontor-Programme ohne langes Suchen aufrufen. Viele Accessories erleichtern Ihnen die Arbeit: RAMDisk, Taschenrechner, Terminkalender, ASCII-Tabelle, Notizblock usw. Die Diskettenverwaltung ersetzt die GEM-Benutzeroberfläche und bietet ihnen viele zusätzliche Möglichkeiten, wie: Formatieren und Kopieren von Disketten, Bearbeiten von Ordnern und Dateien, Suchen von Dateien, Restaurieren gelöschter Dateien, Druckereinstellung. Ein Programm, mit dem Sie Ihre tägliche Arbeit einfacher und effizienter gestalten.

Best.-Nr. 3428, DM 98,–*/sFr. 98,–/S 872,–

Und das kann die ST-Kontor Kundenverwaltung:

Als Basismodul der ST-Kontor-Reihe ist die Kundenverwaltung für die Verwaltung Ihrer Adressen-Bestände zuständig; nur Ihre Festplatte bzw. das RAM können dabei Grenzen setzen. Die wichtigsten Features im Überblick:

- Superschnelles Suchen (Suchen nach Von-Bis-Werten in max. 0,5 Sekunden)
- Optimierte Datenspeicherung
- Mehrere Datenfelder pro Adresse (Stammdaten, Zusatzdaten, Notizen u. a.)
- Umfangreiche Selektionsmöglichkeiten nach beliebig vielen Kriterien
- Ausgabe auf Datei/Monitor/Drucker als Liste, auf Etikett, als Serienbrief

Best.-Nr. 3429, DM 149,–*/sFr. 149,–/S 1326,–

Ein Bündel arbeitssparender Features enthält die ST-Kontor Lagerverwaltung & Fakturierung:

Mit diesem starken Kombi-Programm bringen Sie Ihr Sortiment auf Vordermann: Stücklisten, Etiketten, Preislisten, Kataloge und umfangreiche Angebote mit Textbausteinen sind jetzt kein Problem mehr für Sie. Ebenso selbstverständlich sind verschiedene Lagerbestands-Bewertungsmethoden, Übersicht über den aktuellen Lagerbestand mit Bestellvorschlägen, Fakturierungsübersicht mit Offenen Posten, Teilen und Zusammenlegen von Fakturierungen – und die Übernahme in die Finanzbuchhaltung. Nicht zu vergessen die Bildschirmkasse mit dem aktuellen Kassenstand, Anzeige des Wechselgeldes und Ausdruck eines Kaufbelegs. Und betriebswirtschaftliche Funktionen zur Betriebsoptimierung werden direkt mitgeliefert.

Best.-Nr. 3430, DM 398,–*/sFr. 398,–/S 3542,–

Den aktuellen EG-Richtlinien entspricht die ST-Kontor Finanzbuchhaltung:

Die mehrfirmen- und mandantenfähige FiBu verdaut große Buchungsmengen mit einer erstaunlichen Geschwindigkeit. Unter anderem können Sie von ihr erwarten:

- Kontenrahmen SKR 04 mit 6000 Konten + Einzelhandels-Kontenrahmen SKR 02
- Verschiedene Bilanzierungs- und Abschlußmöglichkeiten mit einfacher Kostenstellen-Rechnung, Anlagenspiegel, Abschreibungs-Verzeichnis und betriebswirtschaftlicher Auswertung
- Verbuchen von Sammelbelegen mit maximal 10 Soll- und 10 Habenkonten sowie 4 Mehrwertsteuer-Sätzen pro Buchung
- Abwicklung des Zahlungsverkehrs mit Offenen Posten und Mahnwesen
- Übernahmedatei für Fakturierung/Lohn & Gehalt mit Korrekturmöglichkeiten

Best.-Nr. 3431, DM 498,–*/sFr. 498,–/S 4432,–

Das erledigt ST-Kontor Lohn & Gehalt für Sie:

In Vorbereitung

- Verwaltung der Mitarbeiterdaten samt Ausfüllen aller Versicherungs- und Finanzamtsformulare
- Alle Lohn- und Gehaltskonten auf einen Blick
- Verwaltung von Zeitkonten einschließlich Krankheits- und Urlaubstagen
- Abrechnung von Vermögensbildung, Direktversicherung und Lohnpfändung
- Komplette Erstellung der Lohnsteuer-Jahresdaten ohne Rechenaufwand
- Abwickeln des Zahlungsverkehrs mit Mitarbeitern, Ämtern und Versicherungen inklusive Berechnung der Zahlungsbeträge und Vorbereitung der Buchung. Die Daten können von der ST-Kontor FiBu übernommen und weiter bearbeitet werden.

Best.-Nr. 3432, DM 198,–*/sFr. 198,–/S 1760,–

*Unverbindliche Preisempfehlung

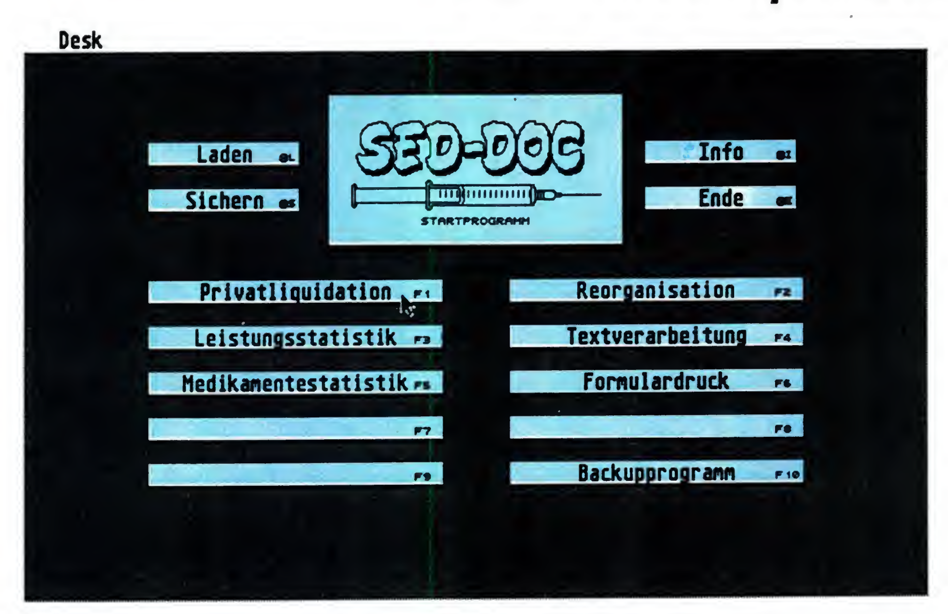
Übrigens: SYBEX sucht ständig gute Buch- und Software-Autoren. Interessiert? Dann kontakten Sie bitte Ralf Lieder.

SYBEX – die guten Seiten Ihres Computers

Sybex Verlag GmbH
Vogelsanger Weg 111
4000 Düsseldorf 30
Telefon: 0211/618020

SED-DOC ST

Eine Privatliquidation



Wenn man als Arzt das Bedürfnis hat, auch technisch 'up to date' zu sein, wird man früher oder später ein elektronisches Abrechnungssystem anschaffen, denn die erforderlichen Aufwendungen machen sich schnell bezahlt. Ich bin niedergelassener Frauenarzt und habe mich nach einigen Vergleichen für das SED-DOC System entschieden. Meine Meinung und einiges über meine Erfahrungen mit dem Programm finden Sie im folgenden Artikel.

Eine 'Privatliquidation'

Für den Laien möchte ich kurz den Begriff Privatliquidation erläutern: Für die erforderlichen Behandlungsmaßnahmen werden Honorare berechnet.
Das Privatliquidationsprogramm verwaltet die Patientendaten, merkt sich die erfolgten Behandlungen, hilft bei der Erstellung von Rechnungen für die Privatpatienten und kümmert sich um das Rechnungschreiben und das Mahnungswesen. Außerdem werden alle Daten der Patienten verwaltet. Somit wird die Arzthelferin entlastet und kann andere Tätigkeiten übernehmen.

Erste Schritte mit SED-DOC

Für mich als Arzt ist der Computer in der Praxis etwas Neues, aber auch Interessantes, so daß ich auch selbst manchmal die Bedienung übernehme:
Zunächst muß vom Desktop ein Startprogramm gestartet werden, das zum Aufruf der Programmodule von SED-DOC (Schnelle Elektronische Datenverarbeitung für DOCtoren) dient. Dieses Startprogramm kann auch als automatisch startend installiert werden (Rechner einschalten und los geht's...).
Von diesem Startprogramm aus wird nun die Leistungsstatistik mit der Maus oder durch eine Funktionstaste aufgerufen. Nachdem man dies getan hat, landet man in der Privatliquidation, die auch ohne das Startprogramm ausgeführt werden kann. Hier wird zunächst überprüft, ob alle Daten einwandfrei sind (dazu später). Außerdem verlangt das Programm eine Bestätigung des Datums, das für die Verwaltung der Patienten, Rechnungen etc. wichtig ist. Angenehm fällt schon hier auf, das man sich nicht um einenKopierschutz kümmern muß, denn der Kopierschutz dieses Programms ist der festinstallierte Name des Arztes. Um diesen ändern zu können, benötigt man eine Kennzahl, die bei Bedarf telefonisch von der Firma mitgeteilt wird.

Eingaben...

Der normale Arbeitsablauf der Rechnungsschreibung wird in der Menüzeile mit dem Eintrag 'Rechnung schreiben' ausgelöst. Daraufhin erscheint ein Patientenformular, das bei der Suche des Patienten/der Patientin hilft. In diesem Formular braucht man nicht einmal den kompletten Namen einzugeben, schon ein Teil des Nachnamens reicht und das Programm stellt alle Patientendaten dar. Nachdem der Patient ausgewählt wurde, geht das Programm in das Rechnungsformular über. Dort wird die Diagnose eingegeben, wobei auch hier optional unter fertigen Diagnosen gewählt werden kann, die dann individuell angepaßt werden können. Als nächstes müssen die Behandlungsmaßnahmen eingegeben werden. Zu jeder Behandlungsmaßnahme können dann die Anzahl und ein Faktor ausgewählt werden (z. B. f. Härtefälle). Gleichzeitig wird automatisch der Rechnungsendbetrag angezeigt. Die Eingabe der Behandlungsmaßnahmen geschieht durch Eingabe der Kennziffern. Sollte eine Ziffer dem Programm nicht bekannt sein (eher eine Ausnahme), kann in einem sodann erscheinenden Formular diese Ziffer neu angelegt werden. Das Programm erfordert kaum Installation. Es lernt während der Eingabe und kann somit vom ersten Augenblick an voll genutzt werden.

Bild 1: Das Patientenformular...

Angenehm fällt auf, daß alle Formulare eine einheitliche Gestaltung aufweisen. Man findet sich schnell zurecht. Positiv anzumerken ist auch die Möglichkeit, in jedem Formular mit 'Help' eine Hilfe holen oder mit 'Undo' falsche Eingaben rückgängig machen zu können. Hat man alle Behandlungsmaßnahmen eingetragen, wobei man in der Rechnung beliebig hin und herfahren und ändern kann, so erscheint ein neues Formular, in dem man noch zusätzliche Kosten oder auch Rabatte in DM oder Prozent eingeben kann.
Oft auftretende zusätzliche Kosten können hier eingegeben werden. Das Programm merkt sie sich dann und wenn sie berechnet werden sollen, reicht dazu ein Mausklick oder ein Tastendruck zur Aktivierung. Hat man die Eingaben bestätigt, so kann die Rechnung gedruckt werden. Mit dem Ausdruck ist der normale Arbeitsablauf abgeschlossen und die Daten werden in der Mahnungsdatei gespeichert.
Die Dateien des Programmes, die für die Rechnungsschreibung verwandt wurden, können außer während des Rechnungschreibens auch über eigene Menüeinträge modifiziert werden. Das Programm unterscheidet Patienten, Versicherungen, Ziffern, Diagnosen und Mahnungen. Diese kann man sich jeweils als Listen ausdrucken oder anzeigen lassen. Außerdem können alle Einträge individuell ergänzt, ediert oder gelöscht werden. Dies geschieht wie im ganzen Programm über komfortabel gestaltete Formulare, die bei der Eingabe auf Fehler prüfen und darauf aufmerksam machen. Glücklicherweise kann das gesamte Programm auch mit der Tastatur bedient werden, so daß das ewige Hin und Her zwischen Tastatur und Maus erspart bleibt.
Nichtautorisierter Benutzung des Programmes, z. B. Abfrage der Patientendaten kann durch die Verwendung eines Paßwortes vorgebeugt werden, so daß in dieser Hinsicht auch der Datenschutz gewährleistet ist.
Bei den Formularen fällt auf, daß einige Einträge vom Programm noch nicht benutzt werden. So kann z. B. der Arbeitgeber eines Patienten zur Zeit schon mit eingegeben werden, er wird im Programm jedoch noch nicht benötigt. Für die zukünftigen Erweiterungsprogramme wie Rezeptdruck, Kassenabrechnung, universelles Textprogramm für Formulardruck etc. ist jedoch durch diese zusätzlichen Einträge sichergestellt, daß alle eingegebenen Daten übernehmbar sind. Die Daten werden nach Angaben der Entwicklungsfirma von diesen zukünftigen Programmen direkt benutzt.

...und Ausgaben

Nun zu den Ausgaben des Programmes oder anders gesagt: Was bringt das Programm auf's Papier? Da wären zunächst einmal die Rechnungen, deren Text man auch frei ändern kann. Gleichzeitig fügt das Programm automatisch einen frei definierbaren Briefkopf mit dem Namen des Arztes usw. ein. Am Ende der Rechnung werden die Konten für die Überweisung angegeben (oder jeder x-beliebige Text...).
Weiterhin interessant ist die Möglichkeit der automatischen Mahnung. Das Programm verwaltet die offenen und beglichenen Rechnungen und sorgt für die fristgerechte Mahnung. Der Benutzer kann eingeben, nach welcher Zeit die 1., 2., oder 3. Mahnung gesandt werden soll. Weiterhin kann auch der mehr oder weniger freundliche Text dieser Mahnungen frei gestaltet werden. Durch die Funk-

Bild 2a: Aus den Behandlungsmaßnahmen und...

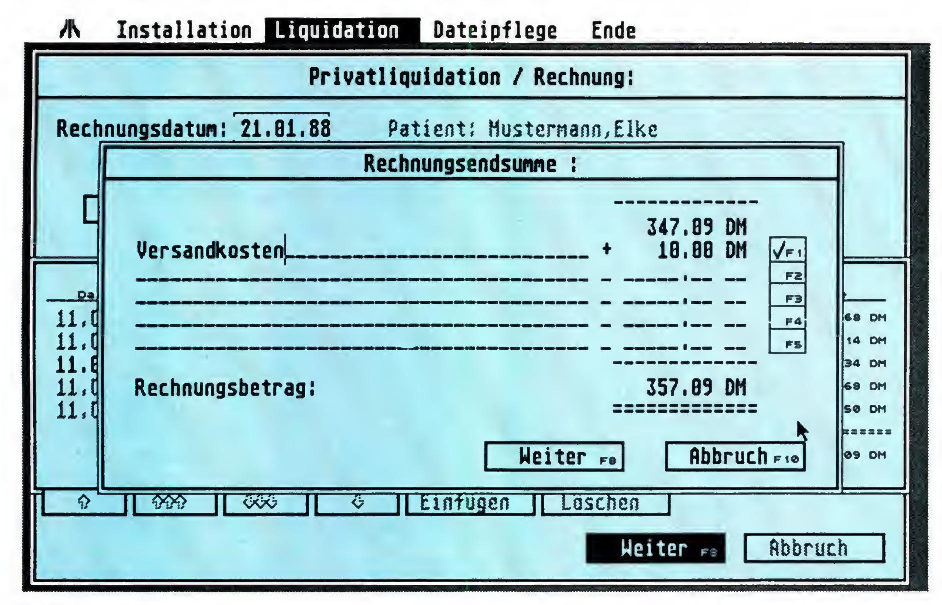


Bild 2b: ...Zusatzkosten...

tionen des Programmes entfällt in der Praxis ein großer Teil stupider und zeitraubender Routinearbeiten. Wenn eine Rechnung beglichen wurde, wird dies dem Programm mit dem Punkt 'Mahnung löschen' rasch mitgeteilt.

Für den Notfall...

Wenn man einmal versehentlich den Rechner ausgeschaltet hat, was natürlichauch durch Stromausfall oder Blitzschlag passieren kann, brauchen Sie sich keine Sorgen um Ihre Daten zu machen. Bei erneutem Aufruf erhält man von der Privatliquidation eine Mitteilung, daß eventuell Daten beschädigt oder nicht vollständig gespeichert wurden. Für diesen Fall ist der Menüpunkt 'Reorganisieren' vorgesehen.
Selbst Dateien, bei denen Fehler durch Hardwaremängel ausgelöst wurden, kann man meist durch das mitgelieferte Reorganisationsprogramm retten. Mit diesem kann ausgewählt werden, welche Datei (Ziffern, Patienten, Diagnosen etc.) reorganisiert werden soll. Sodann zeigt es die Datensätze an und ermöglicht das Löschen, Übernehmen und Korrigieren der Informationen.

Das Handbuch

Ganz im Gegensatz zum Programm weiß ich über das Handbuch nicht viel Positives zu berichten. Glücklicherweise ist die Bedienung des Programms relativ einfach, so daß man das Handbuch selten braucht. Denn wenn man es mal benötigen sollte, wird man seine Zeit brauchen, um die nötige Information zu finden. Die Autoren scheinen noch nichts von einem vernünftigen Inhaltsverzeichnis oder einem Stichwortverzeichnis, geschweige denn von einer Quick-Reference-Tabelle gehört zu haben. Auch sonst scheint insgesamt eine Überarbeitung sehr sinnvoll.
Wie ich jedoch kurz vor Fertigstellung dieses Berichts erfuhr, werde das Handbuch völlig überarbeitet. Das neue Handbuch, bei dessen Gestaltung die Negativerfahrungen eingeflossen seien, soll ab Ende Februar mitgeliefert werden. SED-DOC Anwender erhalten es als kostenloses Update.

Fazit

Ich arbeite bereits mehrere Monate mit dem System und muß sagen, daß ich es heute in meiner Praxis nicht mehr missen möchte. Die Vereinfachung der täglichen Routinearbeit hat sich bezahlt gemacht. Das Programm ist so gestaltet, daß die verschiedensten Fachärzte es benutzen können. Angenehm fällt mir die einfache Bedienbarkeit auf, die ich von elektronischen Datenverarbeitungssystemen nicht erwartet hätte. Der Atari ST leistet hier auch einen Beitrag zur Humanisierung der Arbeit. Besonders schätze ich dabei die vielen Kleinigkeiten, für die der Artikel keinen Platz bietet, die bei der Eingabe nützlich sind und Fehleingaben verhindern. Die Warnungen und Hilfen sind nützlich, um mögliche Eingabefehler zu beheben.
Der Hauptmangel des Programmes ist für mich das Handbuch. Es ist nur zu hoffen, daß die Firma SED-DOC hier in der Zwischenzeit für Abhilfe sorgt.
Der im Vergleich zu sonstigen Softwareprodukten auf dem ST (z. B. Textverarbeitung) hohe Preis ist meiner Meinung nach aufgrund der für Praxiseinsatz erforderlichen Fehlerfreiheit, dem gebotenen Support usw. gerechtfertigt und kann steuerlich schnell abgesetzt werden. Die An-

Installation Liquidation Dateipflege Ende

Ziffernliste : Anzahl: 359

Ziffer	Bezeichnung	GöE	EGO	BMÄ	Faktor
45	Leichenschau	15.20	0.00	0.00	1
4A	Beratung an Samstagen ab 12Uhr	11.40	0.00	0.00	1
4B	Visite im Krankenhaus	6.00	0.00	0.00	1
65	eingehende Untersuchung	10.60	0.00	0.00	1
6A	Besuch aus der Sprechst.heraus	45.00	0.00	0.00	1
70	Schwangerschaftsuntersuchung	24.00	0.00	0.00	1
7A	Nachtbesuch 22-6Uhr-Beratung	62.00	0.00	0.00	1
90	Vorsorgeunters.einer Frau	28.50	0.00	0.00	1
92	Zytologische Untersuchung	13.00	0.00	0.00	1
94	Untersuchung auf Blut im Stuhl	4.20	0.00	0.00	1
9A	Verweilgebühr am Tage	9.50	0.00	0.00	1
9B	Verweilgebühr bei Nacht 20-8	19.00	0.00	0.00	1
10A	Konsilium am Tage am Bett	11.10	0.00	0.00	1
10B	Konsilium b.Nacht am Bett 20-8	23.80	0.00	0.00	1
13A	Assistenz am Tage	24.20	0.00	0.00	1

F1 F2 F3 F4 Suchen: F6 Weiter F9 Abbruch F10

Bild 3: Mit der Listendarstellung...

schaffung des Systems hat sich für mich bereits nach einem halben Jahr amortisiert.
Interessant finde ich außerdem die vom Hersteller angekündigten Erweiterungen. Da diese schon in der jetzigen Version berücksichtigt wurden, kann der Datenbestand problemlos in den neuen Programmen weiterverwandt werden.

Leistungsstatistik

Der Redaktion stand außer der Privatliquidation noch das Programmodul Leistungsstatistik zum Testen zur Verfügung.
Die folgende von der Redaktion verfaßte Kurzbeschreibung soll nur andeuten, was ein solches Programm leistet.
Mit diesem Programm kann der Arzt Tages-, Monats- und Quartalsstatistiken über die von ihm erbrachten Leistungen abrufen. Da bekannt ist, daß die Krankenkassen bei einer Überschreitung der abgerechneten Leistungen Nachforschungen anstellen, kann der Arzt bei Benutzung des Programmes diesen Rahmen besser ausnutzen. Beispielsweise kann er feststellen, daß er bei seinen Behandlungen zu viele Beratungen berechnet hat. In diesem Fall sollte er bei zukünftigen Behandlungen weniger Beratungen veranschlagen, um Ärger mit der Krankenkasse zu vermeiden.
Wie daraus leicht zu erkennen ist, entspricht die Anwendung der Statistikerkenntnisse nicht unbedingt den Wünschen der Krankenkassen.
Ohne EDV ist eine solche Statistik nur schwer durchführbar, d. h. sehr zeitaufwendig. Mit EDV dagegen wird sie spielend einfach, zumal die gesamte Benutzerführung genau wie bei der Privatliquidation gestaltet wurde.
Die besprochene Leistungsstatistik verwendet den EBM (Einheitlicher Bewertungsmaßstab in der Kassenabrechnung vom 1.10.87).

Dr. med H. K. Bönning / jw

Daten von SED-DOC

Speichermöglichkeit:
Disk: 2000 Patienten
Harddisk: 50000 Patienten (empfehlenswert)

Preise:
Leistungsstatistik 598,-
Privatliquidation 898,-
Demo inkl. Handbuch 65,-
(Demo wird beim Kauf angerechnet)

Kontaktadresse:
SED-DOC TEAM
Postfach 4085
5810 Witten 1
Tel.: 02961/50170

ENDE

PADERCOMP — Walter Ladz

Breslauer Straße 25 · 4790 Paderborn · Tel. (0 52 51) 77 07

Floppystationen

PADERCOMP FL 1 3,5", 1 MB, NEC FD1037 A Laufwerk, anschlußfertig, graues Metallgehäuse, Abm.: 153 x 103 x 26 mm	348.00
PADERCOMP FL 2 Doppellaufwerk übereinander, sonst wie FL 1	598.00
PADERCOMP FL 3 Zweitlaufwerk für Amiga	348.00
NEC FD 1037 A 3,5", 1 MB, 25,4 mm Bauhöhe, 5V	225.00
Steckernetzteil 5V/1A, ideal für 2 NEC FD 1037 A	49.00
Industrie-Floppystecker	7.90
ST-Kabel an Shugart-Bus 3,5"	29.90

Drucker

NEC P6 24 Nadeln, 216 Z/s, DIN A4	1098.–
NEC P7 24 Nadeln, 216 Z/s, DIN A3	1448.–
NEC P2200 24 Nadeln, P6-kompatibel	948.–
Star LC-10 9 Nadeln, 144 Z/s, DIN A4	598.–
EPSON LQ-850 24 Nadeln, 264 Z/s, DIN A4	1398.–
EPSON LQ-500 24 Nadeln, 180 Z/s, DIN A4	948.–

Zubehör

FL-1-Adapter zum Anschluß von 2 FL-1	49.00
Multimatic Cut Sheet Feeder für P6	598.00
Dataphon S 21d-2 300 Baud, FTZ-Nr.	238.00
Dataphon S 21/23 300 bzw. 1200/75 Baud, BTX	319.00
Disk Box SS-50 für 50 Stück 3.5"-Disketten	19.90
Media Box 1 für 150 Stück 3,5"-Disketten	34.90
Druckerkabel ST	34.90
Monitor-Ständer dreh-, schwenk- und kippbar	25.00
Monitor-Stecker für ATARI ST	6.90
Preisliste	kostenlos

Monitore

EIZO Flexscan 8060 S alle Auflösungen, kein Nachjustieren, 820 x 620 Punkte, 0,28 dot pitch	1498.–
NEC Multisync alle drei Auflösungen	1398.–
Mitsubishi Freescan Modell 1471	1448.–
Kabel EIZO/NEC an ATARI ST	75.–
Monitor-Umschaltbox	47.–

WIR SIND UMGEZOGEN

Die aktuellen Preise erfragen Sie unter der PCB-Mailbox Tel. 0 52 52-34 13 (300 Bd., 8N1)

Bestellungen per Nachnahme ab 30 DM, Auslandslieferungen nur gegen Vorkasse. Eingetragenes Warenzeichen: ATARI ST. Die Preise können günstiger liegen. Rufen Sie an!

Händleranfragen erwünscht!

ALGORITHMEN & DATENSTRUKTUREN TEIL 7

Steuerspeicherung

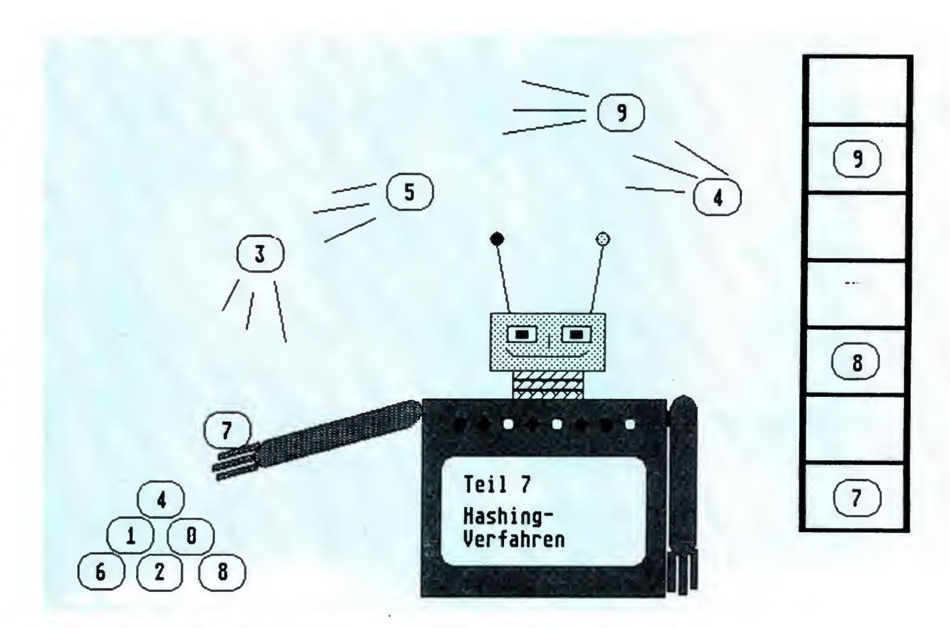
Nachdem wir den Bereich der Bäume in der letzten Folge endgültig verlassen haben, möchte ich Ihnen heute einige Hashing-Verfahren vorstellen.

Begriffe

Dazu gilt es zuvor den Begriff des **Hashing** genauer zu erläutern. Gehen wir davon aus, daß uns eine Tabelle (**Hashingtabelle**) zur Verfügung steht, die wir zur Informationsspeicherung nutzen möchten. Einer solchen Tabelle, die in Pascal üblicherweise in ARRAY-Notation beschrieben wird, liegt nun immer ein Index zugrunde, über den auf die einzelnen Komponenten zugegriffen werden kann.

Beim Hashing, zu deutsch **Streuspeicherung**, gibt man nun eine Funktion (**Hashfunktion**) vor, die eine Abbildung von den Datenschlüsseln in den Index der Tabelle beschreibt. Auf diese Weise wird jedem Schlüssel ein Platz in der Tabelle zugewiesen. Daraus resultiert auch der Begriff **Schlüsseltransformation**, der ebenfalls synonym mit Hashing verwendet wird.

Wesentlich für die Güte der Informationsverarbeitung mit Hashingtabellen ist dabei die Hashfunktion. Es sollte bei der Wahl der Hashfunktion darauf geachtet werden, daß man eine möglichst gleichmäßige Streuung der Daten über die Hashingtabelle erreicht. Unter der Voraussetzung, daß die Schlüssel in etwa gleich verteilt sind und eine Ordinalfunktion (ord) auf den Schlüsseln definiert ist oder definiert werden kann, läßt sich als sehr günstige Hashfunktion folgende Anweisung notieren:




```
hash(key):=abs(ord(key))
MOD tabellenlänge
```

Da diese Funktion Werte im Bereich 0..tabellenlänge-1 annimmt, ist der Index der Hashingtabelle entsprechend zu wählen (siehe auch Listing 7f). Aus weiter unten erläuterten Gründen, sollte die Tabellenlänge eine Primzahl (prim) sein. Der Indexbereich würde dann von 0..prim-1 (primm1) reichen.

Anmerkung:

Anliegend erhalten Sie ein, allerdings nicht näher erläutertes, Programm zur Generierung von Primzahlen (Listing 7g). Nach der Eingabe einer beliebigen Integergröße meldet sich dieses Programm mit der zugehörigen, nächstgrößeren Primzahl, die Sie dann als Tabellengrenze übernehmen können.

Kollisionsbehandlung

Die spezielle Art der Informationsspeicherung mit Hashingtabellen wird natürlich nur dann sinnvoll, wenn die Anzahl der möglichen Schlüssel, zumeist in hohem Maße, die Anzahl der Tabellenplätze über-

steigt. Andernfalls wäre es wesentlich sinnvoller die Schlüsselmenge, oder eine einfache Ordinalmenge davon, als Indexmenge des Arrays zu benutzen und die Daten direkt abzuspeichern.

Wenn wir aber davon ausgehen, daß die mögliche Anzahl der Schlüssel die Tabellenplätze übersteigt, so wird es zwangsläufig Situationen geben, wo unsere Funktion mehrere Schlüssel auf ein und denselben Tabellenplatz abbildet. Es sind also Verfahrensweisen anzugeben, wie derartige Zusammentreffen, genannt **Kollisionen**, zu behandeln sind.

Generell werden dazu zwei unterschiedliche Strategien angewandt:

Die erste Strategie geht davon aus, daß ein Tabellenplatz nicht nur ein Element, sondern eine ganze Liste von Elementen aufnehmen kann. Geht man nun bei der Listenverwaltung mit dem Mittel der dynamischen Speicherverwaltung vor, so entstehen keine weiteren Probleme, da nun beliebig viele Elemente auf einen Tabellenplatz abgebildet werden können. Dieses Verfahren wird offene Streuspeicherung genannt.

Bei dem entgegengesetzten Ansatz, der geschlossenen Streuspeicherung, wird versucht, einen Ersatzplatz für eine belegte Stelle der Hashingtabelle zu finden. Dabei kommt es zum sogenannten Rehashing, indem, als Funktion der Tabellenposition, eine Ersatzstelle berechnet wird.

Geschlossene Streuspeicherung

Befassen wir uns zunächst mit dem letztgenannten Ansatz, der geschlossenen Streuspeicherung.

Die dabei notwendigen Typdefinitionen finden Sie im Listing 7a. Die Hashtabelle enthält dabei jeweils Einträge (item_type), die sich aus Schlüssel (key), Daten (data), und einem zusätzlichen Flag (used), zur Markierung der belegten/unbelegten Tabelleneinträge, zusammensetzen. Diese Tabelle ist, wie Sie sicherlich bemerkt haben, nicht dynamisch ! Das ist auch DER große Nachteil des Verfahrens der geschlossenen Streuspeicherung: Von vornherein liegt die Obergrenze der Tabelle fest.

Ein weiteres Problem liegt in der Wahl einer geeigneten Rehashingfunktion.

Die einfachste mögliche Rehashingfunktion, die man zur Behandlung der Kollisionen angeben kann, ist das Überprüfen der nächsten Stelle in der Hashingtabelle. Dabei geht man, wie bei allen folgenden Überlegungen, davon aus, daß die Tabelle als Ring aufgefasst werden kann und alle Indizes somit modulo Tabellenlänge zu rechnen sind. Mithin ist es auch möglich, die Nachfolger für Positionen am Tabellenende zu bestimmen. Führt das Rehashing wiederum zu keinem Ergebnis, so ist es solange durchzuführen, bis ein unbenutzter Tabellenplatz gefunden wurde, oder eine gewisse Abbruchbedingung erreicht wird. Im letzten Fall kann kein Einfügen vorgenommen werden.

Die oben beschriebene Vorgehensweise - Wahl des Tabellennachfolgers als Rehashingergebnis - nennt man **lineares Sondieren**. Obwohl Sie die einfachste mögliche Rehashingfunktion ist, ist die Verwendung nicht unbedingt empfehlenswert. Beim linearen Sondieren neigt die Tabelle nämlich dazu, größere Gruppen von belegten Plätzen zu bilden. Wird nämlich einmal ein Rehashingvorgang eingeleitet, so besitzt man schon eine Gruppe von zwei aufeinanderfolgenden Tabellenpositionen. Beim nächsten Treffer dieser Gruppe, der wegen wachsender Größe immer wahrscheinlicher wird, sind schon maximal zwei Rehashingschritte notwendig ...

Es kommt also zu größeren Gruppen von Elementen, die bei Einfüge- und Suchvorgängen in der Regel einige Rehashingschritte notwendig machen.

Ausweg aus dieser Situation ist die Wahl einer Rehashingfunktion, die größere Schrittweiten benutzt. Die nächst'höhere' Funktion nach dem linearen Sondieren ist das **quadratische Sondieren**.

Hierbei wird nicht ein linear ansteigender Wert (1,2,3,4,...) zur Ausgangsposition addiert, sondern ein quadratisch ansteigender Wert (1,4,9,16,...). Die dabei lauernde Gefahr besteht in der Möglichkeit, beim Rehashing nicht alle Tabellenplätze zu erreichen und somit eventuell ein Element nicht einfügen zu können, obwohl in der Tabelle noch Plätze zu vergeben sind. Dies ist auch der Grund, weswegen man als Tabellengröße gerade eine Primzahl benutzt, weil diese Begebenheit bei der Restklassenbildung über die Quadrate gewährleistet, daß nicht unnötig viele Tabellenplätze doppelt inspiziert werden (Die Mathematik läßt grüssen !). Trotzdem kommt es bei starkem Füllgrad der Hashingtabellen und quadratischem Sondieren manchmal zu dem Fall, daß trotz noch vorhande-

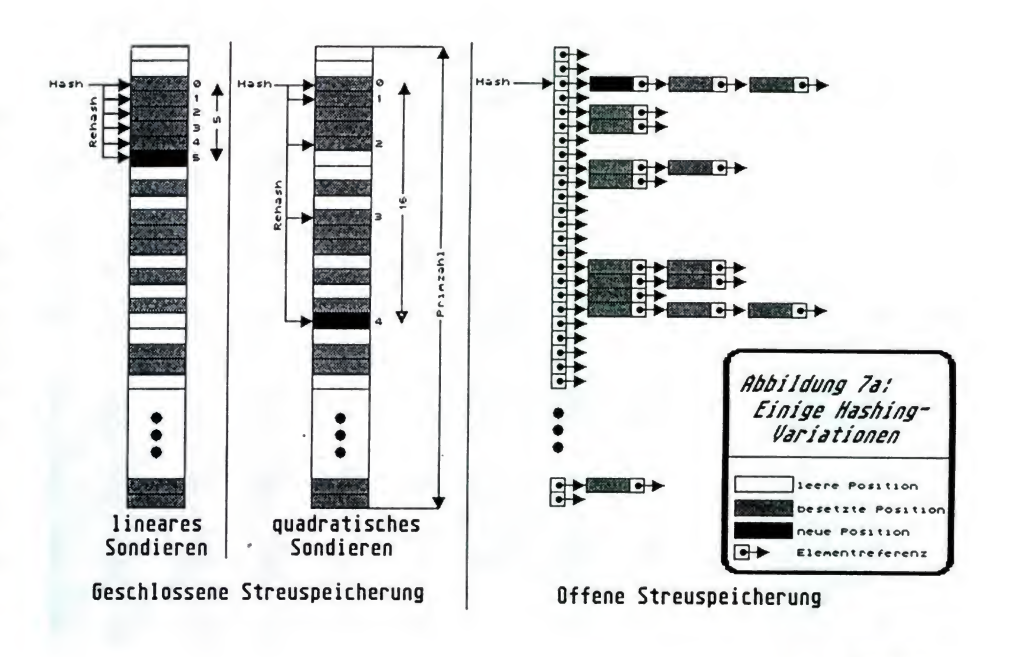


Abbildung 7a: Einige Hashing-Variationen

nem Platz die Einfügestelle nicht gefunden wird. Zu Ihrem Trost sei jedoch gesagt, daß dies nur in Größenordnungen von 98-100% Füllgrad auftritt, also prinzipiell vernachlässigt werden kann.
Eine weitere Verbesserung erreicht man durch die Benutzung einer rekursiven Formel zur Erzeugung der beim Sondieren notwendigen Quadratberechnungen. Diese Formel hat folgendes Aussehen:

```
rehash:=rehash+d;
d:=d+2;
```

So ergibt sich, mit Startwert für d:=1 und rehash:=hash(key), zusätzlich zur einfacheren Berechnung der Quadrate sogar ein bequemes Abbruchkriterium (d=prim) für das Rehashing.
Wie diese einzelnen Rehashing-Methoden zu implementieren sind, können sie den Listings 7b (lineares Sondieren) und 7c (normale und rekursive Variante des quadratischen Sondierens) entnehmen.

Einfügen und Suche bei geschlossener Streuspeicherung

Das Einfügen (insert) gestaltet sich mit den beiden vorgegebenen Funktionen hash und rehash (Listing 7b) sehr einfach. Zunächst wird der einem Schlüssel zugehörige Tabellenwert ermittelt. Handelt es sich dabei um einen belegten Tabellenplatz, so ist mit rehash eine Ersatzstelle zu ermitteln. Konnte keine Ersatzstelle ermittelt werden, gehen wir davon aus, daß uns unsere rehash-Funktion als Ergebnis die alte Position zurückliefert. In diesem Fall ist die Variable found entsprechend zu setzen. Abschließend ist nur noch, im Falle found, der entsprechende Tabellenplatz zu übergeben und insert der Erfolg oder Mißerfolg des Einfügens durch Zuweisung von found mitzuteilen.
Die Suche (search) verläuft ähnlich. Auch hier ist zunächst mit hash eine Einfügestelle zu ermitteln. Wenn diese nicht leer ist und der Schlüssel auch noch nicht gefunden ist (Zeile 71), so ist mit rehash weiterzusuchen, bis entweder ein leerer Tabellenplatz oder ein Tabellenplatz mit bezeichnetem Schlüssel gefunden wurde. In diesem Fall sind die Daten zuzuweisen.

Offene Streuspeicherung

Bei der offenen Streuspeicherung wird nun ein etwas verändertes Typkonzept benutzt (Listing 7d). Die Tabelle wird nun nicht mehr als Aufbewarungsort für die Schlüssel und Daten benutzt, sie dient lediglich der Referenz auf den Anfang der Eintragsliste (ptr_type). Diese Tabelleorganisation macht es uns sehr leicht, ein Kriterium für einen leeren Tabellenplatz zu wählen: Wir nehmen dazu einfach die Zeigerkonstante nil. Die Einträge (item_type) sind nun, entsprechend Listenmanier, noch mit einem Zeiger next versehen. Dafür fehlt, im Gegensatz zum vorherigen Konzept, das benutzt-Flag.

Einfügen und Suchen bei offener Streuspeicherung

Kollisionsbehandlung entfällt bei diesem Konzept vollständig. Ist einmal, mit einer Hashfunktion, eine Einfügestelle ermittelt worden, so ist das neue Element lediglich zu Beginn der Liste einzuhängen. Die alte Liste, bei leeren Listen nur der Zeiger nil, wird dabei als Nachfolger des neuen Elementes eingetragen. Dadurch ist auch bereits der komplette Einfügevorgang beschrieben (Listing 7e, Zeilen 23-36). Bei der Suche (ebenfalls 7e, 38-58) ist nun zu beachten, daß es sich bei der durch das Einfügen erzeugten Liste um eine unsortierte Liste handelt. Deshalb ist die komplette Liste bis zum gesuchten Element zu durchlaufen. Auf eine kleine Inkonsistenz möchte ich Sie noch hinweisen: Doppelt auftretende Tabellenelemente können Sie auf diese Weise nicht auslesen. Benutzen Sie dazu am besten die search_first - search_next Taktik, die Ihnen aus den letzten Folgen sicherlich hinreichend vertraut ist.

Verbesserungsvorschläge

Prinzipiell ist es bei offener Streuspeicherung auch möglich, Elemente aus der Hashingtabelle zu löschen, da dies einem reinen Listenausfügen entspricht. Von dieser Möglichkeit habe ich aber im Beispiel keinen Gebrauch gemacht. Dies sei ebenfalls dem interessierten Leser überlassen.
Im Gegensatz zur offenen Streuspeicherung wird man bei der geschlossenen Streuspeicherung vergeblich nach einer Möglichkeit suchen, Elemente zu löschen. Dies ist durch die Tatsache bedingt, daß durch das Ausfügen eines Elementes in den Rehashingstufen anderer Suchvorgänge eine Lücke entsteht. Dadurch besteht kein eindeutiges Abbruchkriterium für das Rehashing (leeres Tabellenelement) mehr.
Einen weiteren, recht einfach zu realisierenden Verbesserungsvorschlag möchte ich noch nennen.
Gehen wir einmal davon aus, daß die Tabelle, im Vergleich mit der auftretenden Anzahl von Elementen, sehr klein ist. In diesem Fall wird die Liste recht lang werden und es empfiehlt sich die Nutzung einer höheren Struktur, zum Beispiel die eines **AVL-Baumes**, zur Organisation der eintreffenden Elemente. Eventuell - ausprobiert habe ich es noch nicht - ergibt sich durch die Kombination dieser beiden unterschiedlichen Konzepte eine Suchstruktur, die nocheinmal wesentlich bessere Suchwege als die Suchbäume der vorhergehenden Folgen aufweist.

Messungen

Was ich allerdings ausgetestet habe, ist die Effizienz der einzelnen Hashingmethoden.
Als Kriterium für die Güte bei der geschlossenen Streuspeicherung habe ich die durchschnittliche Anzahl der einzelnen Rehashingschritte beim Einfügen, bis zur Erreichung eines gewissen Füllungsgrades, ermittelt (Abbildung 7b).

Abbildung 7b: Leistungsanalyse Hashingverfahren

% Anteil	Geschlossene Streuspeicherung		Offene Streuspeicherung
	lineares Sondieren	quadratisches Sondieren	
	Anzahl der Rehashingschritte		Tabellenfüllungsgrad
50	0.51	0.43	0.40
70	1.16	0.87	0.51
80	1.90	1.13	0.55
90	3.88	1.77	0.61
95	6.13	2.06	0.61
97	9.22	2.71	0.63
98	13.22	3.18	0.63
99	16.35	3.88	0.63
100	21.73	(5.02)	0.64

Dabei stellte sich heraus, daß beide Sondierungen unterhalb 90% Füllung sehr gute Werte erreichen (lineares unterhalb 4 Rehashingschritte; quadratisches unterhalb 2 Rehashingschritte). Besonders das quadratische Sondieren setzt seine gegenüber dem linearen Sondieren ohnehin positive Bilanz auch im Bereich von 90-100% fort. Der letzte Eintrag des quadratischen Sondierens - für einen Tabellenfüllgrad von 100% - ist allerdings mit Vorsicht zu genießen, da hier nicht mehr bei allen Versuchen die komplette Anzahl an Elementen eingefügt werden konnte.

Als Gütekriterium für die offene Streuspeicherung wurde der Tabellenfüllungsgrad, also die Anzahl der besetzten im Vergleich zu den unbesetzten Listen, ermittelt. Die Ergebnisse sind auch hier recht positiv, da bei gleichgroßem Verhältnis von Tabellenplätzen zu Elementen durchschnittlich 64% der Listenplätze vergeben waren. Dies bedeutet eine durchschnittliche Listenlänge von weit unter zwei! Die Suchzeiten dürften also, bei nicht allzugroßer Überfüllung, noch unterhalb derjenigen der geschlossenen Streuspeicherung liegen. Ganz abgesehen von dem Vorteil, den man durch die dynamische Struktur erreicht.

Die daraus resultierenden Anwendungsempfehlungen gehen in zwei Richtungen:

A: Bei von vornherein bekannter Größe des Datenaufkommens und einem beschränkten Speicher - beispielsweise bei der Anwendung von Hashing Tabellen in Accessories -, sollte man die geschlossene Streuspeicherung, mit quadratischem Sondieren, wählen. Die Tabellengröße ist dabei so vorzugeben, daß bei maximalem Füllgrad etwa 90-95% belegt sind.

B: Alle anderen Anwendungen verlangen nach der offenen Streuspeicherung, eventuell mit einer höheren Struktur zur Sekundärschlüsselverwaltung.

Testumgebung

Die Testumgebung (Listing 7f) erlaubt nun, entweder von Hand oder zufällig und in größeren Mengen, Elemente in die Hashingtabelle aufzunehmen bzw. nach Elementen zu suchen. Sie ist geschrieben für die geschlossene Streuspeicherung. Bei der Umwandlung in eine Testumgebung der offenen Streuspeicherung beachten Sie bitte die im Programmkopf gegebenen Hinweise.

Vorausschau

In der nächsten Folge der Algorithmen & Datenstrukturen verlassen wir endgültig den Bereich der Informationsstrukturen. Das nächste und letzte größere Thema lautet Sortieralgorithmen. Hierbei werde ich Ihnen in den letzten drei Teilen dieser Serie, angefangen mit den einfachen Sortieralgorithmen bis hin zu den klassischen Algorithmen wie Shellsort, Heapsort und Quicksort, die gebräuchlichsten Formen des Sortierens vorstellen.

Sig

```
 0: { Listing 7a   : Typen einer Hashing-Tabelle mit
 1:                  geschlossener Streuspeicherung.
 2:
 3:   programmiert : November 87
 4:   von          : Dirk Brockhaus
 5:   mit          : PASCAL+ (CCD)
 6: }
 7:
 8:       key_type  = integer;
 9:
10:       data_type  = integer;
11:
12:       item_type  = RECORD
13:                      used : boolean;
14:                      key  : key_type;
15:                      data : data_type;
16:                    END;
17:
18:       hash_range = 0..primm1;
19:
20:       hash_table = ARRAY [hash_range] OF item_type;
```

```
 0: { Listing 7b   : Unterroutinen zu den Hashing-Tabellen
 1:                  bei geschlossener Streuspeicherung.
 2:
 3:   programmiert : November 87
 4:   von          : Dirk Brockhaus
 5:   mit          : PASCAL+ (CCD)
 6: }
 7:
 8: PROCEDURE init_table;
 9:
10:   VAR i : hash_range;
11:
12:   BEGIN {init_table}
13:     FOR i:=0 TO primm1 DO
14:       table[i].used:=false;
15:   END;  {init_table}
16:
17: FUNCTION hash (key : key_type) : hash_range;
18:
19:   BEGIN {hash}
20:     hash:=abs(key) MOD prim;
21:   END;  {hash}
22:
23: FUNCTION rehash {linear}(pos : hash_range) : hash_range;
```

```
24:
25:   VAR i : integer;
26:
27:   BEGIN {rehash}
28:     i:=pos+1;
29:     WHILE (table[i MOD prim].used) AND
30:           (table[i MOD prim].key<>key) AND
31:           (i MOD prim <>pos) DO
32:       i:=i+1;
33:     rehash:=i MOD prim;
34:   END;  {rehash}
35:
36: FUNCTION insert (key  : key_type;
37:                  data : data_type) : boolean;
38:
39:   VAR pos     ,
40:       new_pos : hash_range;
41:       found   : boolean;
42:
43:   BEGIN {insert}
44:     pos:=hash(key);
45:     IF table[pos].used THEN
46:       BEGIN
47:         new_pos:=rehash(pos);
48:         found:=new_pos<>pos;
49:         pos:=new_pos;
50:       END
51:     ELSE
```

```
52:       found:=true;
53:     IF found THEN
54:       BEGIN
55:         table[pos].used:=true;
56:         table[pos].key:=key;
57:         table[pos].data:=data;
58:       END;
59:     insert:=found;
60:   END;  {insert}
61:
62: FUNCTION search (     key  : key_type;
63:                   VAR data : data_type) : boolean;
64:
65:   VAR pos   : hash_range;
66:       found : boolean;
67:
68:   BEGIN {search}
69:     pos:=hash(key);
70:     found:=(table[pos].key=key) AND table[pos].used;
71:     IF NOT found THEN
72:       BEGIN
73:         pos:=rehash(pos);
74:         found:=(table[pos].key=key) AND table[pos].used;
75:       END;
76:     IF found THEN
77:       data:=table[pos].data;
78:     search:=found;
79:   END;  {search}
```

```
0: { Listing 7c    : Zwei Variationen zum quadratischen
                     Sondieren.
1:   Anmerkung     : Dieses Sondieren führt ziemlich schnell
2:                   zu sehr großen Werten in den lokalen
3:                   Integergrößen, so daß bereits bei
4:                   Tabellen unter 1000 Elementen im
                     'normalen' Integerbereich Überlauf
                     stattfindet.
5:
6:   programmiert  : November 87
7:   von           : Dirk Brockhaus
8:   mit           : Pascal+ (CCD)
9: }
10:
11: FUNCTION rehash {quadratisch} (pos : hash_range)
12:                                    : hash_range;
13:   VAR i ,
14:       d : integer;
15:
16:   BEGIN {rehash}
17:     i:=pos;
18:     d:=1;
19:     REPEAT
20:       i:=pos+d*d;
21:       d:=d+1;
22:     UNTIL (NOT table[i MOD prim].used) OR
```

```
23:           (table[i MOD prim].key=key) OR
24:           (d*d>=prim);
25:     IF d*d>=prim THEN
26:       rehash:=pos
27:     ELSE
28:       rehash:=i MOD prim;
29:   END;  {rehash}
30:
31: FUNCTION rehash {quadratisch mit Rekursion}
32:                 (pos : hash_range) : hash_range;
33:
34:   VAR i ,
35:       d : integer;
36:
37:   BEGIN {rehash}
38:     i:=pos;
39:     d:=1;
40:     REPEAT
41:       i:=i+d;
42:       d:=d+2;
43:     UNTIL (NOT table[i MOD prim].used) OR
44:           (table[i MOD prim].key=key) OR
45:           (d=prim);
46:     IF d=prim THEN
47:       rehash:=pos
48:     ELSE
49:       rehash:=i MOD prim;
50:   END;  {rehash}
```

```
0: { Listing 7d    : Typen einer Hashing-Tabelle mit offener
1:                   Streuspeicherung.
2:
3:   programmiert  : November 87
4:   von           : Dirk Brockhaus
5:   mit           : PASCAL+ (CCD)
6: }
7:
8:      key_type   = integer;
9:
10:     data_type  = integer;
11:
```

```
12:      ptr_type   = ^item_type;
13:
14:      item_type  = RECORD
15:                     key  : key_type;
16:                     data : data_type;
17:                     next : ptr_type;
18:                   END;
19:
20:      hash_range = 0..primm1;
21:
22:      hash_table = ARRAY [hash_range] OF ptr_type;
```

```
0: { Listing 7e    : Unterroutinen zu den Hashing-Tabellen
1:                   bei offener Streuspeicherung.
2:
3:   programmiert  : November 87
4:   von           : Dirk Brockhaus
5:   mit           : PASCAL+ (CCD)
6: }
7:
8: PROCEDURE init_table;
9:
10:   VAR i : hash_range;
11:
```

```
12:   BEGIN {init_table}
13:     FOR i:=0 TO primm1 DO
14:       table[i]:=nil;
15:   END;  {init_table}
16:
17: FUNCTION hash (key : key_type) : hash_range;
18:
19:   BEGIN {hash}
20:     hash:=abs(key) MOD prim;
21:   END;  {hash}
22:
23: PROCEDURE insert (key  : key_type;
24:                   data : data_type);
```

ATARI ST BUCHHITS!

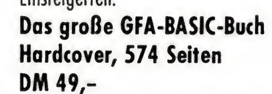


Auspacken und gleich loslegen. Atari ST für Einsteiger macht's möglich. Vom Aufstellen und Anschließen über die Arbeit mit dem GEM-Desktop bis hin zum ST-BASIC – mit diesem Buch haben Sie die optimale Einführung zu Ihrem neuen Rechner. So ist der Erfolg bei Ihrer späteren Arbeit sozusagen schon vorprogrammiert.

ATARI ST für Einsteiger
248 Seiten, DM 29,–



Das meistverkaufte Buch zu GFA-BASIC – und das nicht ohne Grund. Denn hier lernen Sie den kompletten Befehlssatz des GFA-BASIC Version 2.0 und den GFA-Compiler anhand zahlreicher Beispielprogramme kennen. Ganz ohne die sonst übliche, nackte Befehlsübersicht. Das große GFA-BASIC-Buch – natürlich mit einem ausführlichen Einsteigerteil.



Das große GFA-BASIC-Buch
Hardcover, 574 Seiten
DM 49,–



Das Buch zum neuen Super-ST. Einstieg, DTP, Arbeiten mit Laserdrucker und Festplatte, TOS, Blitter, Betriebssystem-Programmierung – was Sie zum MEGA ST wissen müssen, finden Sie hier. Mit kommentiertem TOS-Blitter-Listing. Mit diesem Buch liegt Ihnen die gesamte Mega-Power zu Füßen.

Das große MEGA-ST-Buch
Hardcover, 511 Seiten
inkl. Diskette, DM 69,–

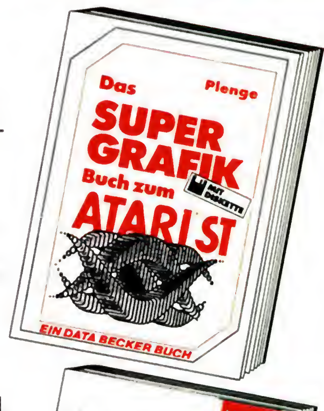


DATA BECKER Führer zum ATARI ST
240 Seiten
DM 29,80



DATA BECKER Führer zu GFA-BASIC
254 Seiten
DM 24,80

Drei Bücher zum ATARI ST ersparen Ihnen das lange Suchen im Handbuch oder in der Fachliteratur. Fehlt Ihnen eine bestimmte Information, können Sie immer ganz gezielt nachschlagen: Die DATA BECKER Führer – die erfolgreichste und kompletteste Serie ihrer Art. Kompetent, zuverlässig und immer griffbereit.



Das Supergrafikbuch zum ST – vollgepackt mit dem Know-how, das jeder engagierte ST-Anwender braucht. Von den Grundlagen bis zu speziellen Problemlösungen wie Programmierung eines Rasterinterrupts oder einer flackerfreien Animation finden Sie hier alles zum Thema Grafik. Mit zahlreichen Utilities in GFA-BASIC, C und Assembler. Ein Buch nicht nur für Grafik-Freaks.

Das Supergrafikbuch zum ATARI ST
Hardcover, 838 Seiten
inkl. Diskette, DM 69,–



Alles zu der Textverarbeitung 1st Word Plus – einschließlich der Zusatzprogramme 1st Mail, 1st Lektor, 1st Proportional und 1st Index. Mit vielen Tips und Lösungen aus der praktischen Arbeit. Nicht nur ein Lehrbuch, sondern auch ein hervorragendes Nachschlagewerk.

Das große Buch zu 1st Word Plus
Hardcover, 288 Seiten
inkl. Diskette, DM 59,–



Intern-Bände von DATA BECKER sind seit jeher Informationspakete ganz besonderer Art. So auch hier: Von der Pinbelegung bis hin zum kommentierten BIOS-Listing wird jedes Detail Ihres Rechners ausführlich beschrieben – natürlich auch der Atari-Blitter. Einfach die Pflichtlektüre für den ST-Profi.

ATARI ST Intern
Hardcover, 637 Seiten
DM 69,–



DATA BECKER Führer zu 1st Word
192 Seiten
DM 24,80

COUPON

HIERMIT BESTELLE ICH

NAME, VORNAME

STRASSE, ORT

zzgl. DM 5,– Versandkosten unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

```
25:
26:   VAR pos     : hash_range;
27:       element : ptr_type;
28:
29:   BEGIN {insert}
30:     pos:=hash(key);
31:     new(element);
32:     element^.next:=table[pos];
33:     element^.data:=data;
34:     element^.key:=key;
35:     table[pos]:=element;
36:   END;  {insert}
37:
38: FUNCTION search (     key  : key_type;
39:                   VAR data : data_type) : boolean;
40:
41:   VAR pos   : hash_range;
42:       work  : ptr_type;
43:       found : boolean;
44:
45:   BEGIN {search}
46:     pos:=hash(key);
47:     work:=table[pos];
48:     found:=false;
49:     WHILE (work<>nil) AND NOT found DO
50:       IF work^.key<>key THEN
51:         work:=work^.next
52:       ELSE
53:         BEGIN
54:           data:=work^.data;
55:           found:=true;
56:         END;
57:     search:=found;
58:   END;  {search}
```

```
 0: { Listing 7f   : Testumgebung zu den Hashing-Tabellen
 1: (sowohl offene, als auch geschlossene Streuspeicherung).
 2:
 3:  Anmerkung : Dieser Source verwaltet eine H-Tabelle mit
 4:              geschlossener Streuspeicherung.
 5:  Zur Umwandlung in eine Testumgebung für die offene
 6:  Streuspeicherung beachten Sie bitte folgende Punkte:
 7:
 8:  A: Ersetzen der Include-Anweisungen
 9:     i g_typ.pas     durch    i o_typ.pas
10:     i g_sub.pas     durch    i o_sub.pas
11:  B: Umformulierung der Aufrufe der Funktion 'insert'
12:     in Prozeduraufrufe, da eine offene Tabelle nicht
13:     überläuft.
14:
15:  Bei Benutzung des quadratischen Sondierens benutzen
16:  Sie bitte die Compileroption I+ vor PROGRAM, um
17:  den Integerbereich auf Long_Integer hochzusetzen.
18:
19:  programmiert : November 87
20:  von          : Dirk Brockhaus
21:  mit          : PASCAL+ (CCD)
22: }
23:
24: PROGRAM hashing (input,output);
25:
26: CONST prim   = 997;
27:       primm1 = 996;
28:
29: TYPE  {$i g_typ.pas}
30:
31: VAR   table : hash_table;
32:       key   : key_type;
33:       data  : data_type;
34:
35:       frage : char;
36:       anz   ,
37:       i     : integer;
38:
39: {$i g_sub.pas}
40:
41: FUNCTION random : integer;
42:   XBIOS(17);
43:
44: BEGIN {main}
45:   init_table;
46:   REPEAT
47:     page(output);
48:     writeln('Test von Hashing-Tabellen');
49:     writeln;
50:     writeln('(1) Einfuegen');
51:     writeln('(2) Suchen');
52:     writeln('(3) Random Einfuegen');
53:     writeln('(4) Tabelle loeschen');
54:     writeln;
55:     writeln('(0) Beenden');
56:     writeln;
57:     write('--->');  read(frage);
58:     writeln;
59:     writeln;
60:     CASE frage OF
61:       '1' : BEGIN
62:               writeln('Element einfuegen');
63:               write('Schluessel ? '); readln(key);
64:               write('Daten ? '); readln(data);
65:               IF NOT insert(key,data) THEN
66:                 writeln('Tabelle voll !
                              ==> Einfuegen nicht moeglich');
67:             END;
68:       '2' : BEGIN
69:               writeln('Schluessel suchen');
70:               write('Schluessel ? '); readln(key);
71:               IF search(key,data) THEN
72:                 writeln('Zugehoerigen Daten: ',data)
73:               ELSE
74:                 writeln('Schluessel nicht gefunden');
75:             END;
76:       '3' : BEGIN
77:               writeln('Random Schluessel einfuegen');
78:               write('Anzahl ? '); readln(anz);
79:               FOR i:=1 TO anz DO
80:                 IF NOT insert(random,random) THEN
81:                   writeln('Tabelle voll');
82:             END;
83:       '4' : init_table;
84:     END;
85:   UNTIL frage='0';
86: END.  {main}
```

```
 0: { Listing 7g   : Tabellenende berechnen
 1:   programmiert : November '87
 2:   von          : Dirk Brockhaus
 3:   mit          : Pascal+ (CCD)
 4: }
 5:
 6: PROGRAM tabellen_grenze_ermitteln (input,output);
 7:
 8:   VAR anz : long_integer;
 9:
10:   FUNCTION primzahl(x : long_integer) : boolean;
11:
12:     VAR i     : long_integer;
13:         found : boolean;
14:
15:     BEGIN {primzahl}
16:       IF x<=1 THEN
17:         primzahl:=false
18:       ELSE
19:         IF x MOD 2=0 THEN
20:           primzahl:=x=2
21:         ELSE
22:           BEGIN
23:             i:=3;
24:             found:=false;
25:             WHILE (i*i<=x) AND NOT found DO
26:               BEGIN
27:                 found:=x MOD i=0;
28:                 i:=i+2;
29:               END;
30:             primzahl:=NOT found;
31:           END;
32:     END;  {primzahl}
33:
34:   PROCEDURE next_prim(VAR start : long_integer);
35:
36:     VAR found : boolean;
37:
38:     BEGIN {next_prim}
39:       IF NOT odd(start) THEN
40:         start:=start+1;
41:       found:=false;
42:       WHILE NOT found DO
43:         BEGIN
44:           found:=primzahl(start);
45:           start:=start+2;
46:         END;
47:       start:=start-2;
48:     END;  {next_prim}
49:
50: BEGIN {main}
51:   write('Minimale Tabellengroesse ? '); readln(anz);
52:   next_prim(anz);
53:   writeln('Empfohlene Tabellengroesse : ',anz);
54:   writeln('Weiter mit <Return>');
55:   readln;
56: END.  {main}
```

SUPERCHARGER

BETA SYSTEMS AG

Der MS - DOS Emulator für Atari ST
- volle PC - Kompatibilität, erweiterbar
- Benutzung der Atari Peripherie
- Maus Unterstützung
- Multitaskingfähig
- Netzwerk für Atari ST Rechner
- 1 MB RAM - auch als Atari RAM - Disk nutzbar

Bei Vorauskasse 3% Skonto **DM 698.-**
Nähere Infos bitte anfordern (DM 2 - Rückporto)

SUPERCHARGER

+ ARC - Floppy 5 1/4" (AF-2)

DM 999.-

ARC - Floppy AF - 5

5 1/4" Diskstation, 40/80 Spur umschaltbar, komplett anschlußfertig, begrenzte Stückzahl nur 100 Stück

Aktionspreis: **DM 349.-**

ARC - Floppy AF - 3

3 1/2" Floppy, doppelseitig, 720 KB incl. Netzteil und Gehäuse

DM 349.-

ARC - Harddisk AD - 20

20 MB Speicherkapazität, Atari-Interface, Zugriffszeit: 65 mS

DM 998.-

ADS - 20: Harddisk AD - 20

mit 20 MB - Streamer **DM 2398.-**

ARC - Harddisk AD - 30

30 MB Speicherkapazität, Atari-Interface, Zugriffszeit: 65 mS

DM 1298.-

ADS - 30: Harddisk AD - 30

incl. Streamer **DM 2698.-**

ARC - Harddisk AD - 40

40 MB Speicherkapazität, Atari-Interface, Zugriffszeit: 28 mS

DM 2198.-

ADS - 40: Harddisk AD - 40

mit 40 MB Streamer **DM 3598.-**

Für Profis:

ARC - Harddisk AD - 80

80 MB Speicherkapazität, Atari-Interface, Zugriffszeit: 28 mS

DM 3598.-

ARC - Akustikkoppler 300

flexibles Mittelteil: deshalb optimale Anpassung an verschiedene Telefonhörer, 300 Band, Stromversorgung 9V, Originate / Answer Modus

DM 189.-

BTX - Manager

Nutzen Sie Ihren ST für die unendlich vielen BTX - Möglichkeiten. Die intelligente Komplettlösung.

DM 399.-



BESTELLCOUPON

- ☐ Senden Sie mir Ihren Katalog (Schutzgebühr DM 3 - in Briefmarken)
- ☐ Senden Sie mir nähere Infos über den Artikel ______
- ☐ Hiermit bestelle ich zu Ihren Liefer- und Geschäftsbedingungen per NN ☐ per Vorauskasse ☐

Bei Aufträgen unter DM 200 zzgl. Versandkosten-Pauschale DM 5,90

Bitte vergessen Sie nicht Ihre Anschrift und Telefonnummer

A 1

07193 / 8699

ARC

Arc Computervertriebs GmbH, Talstraße 16, 7158 Sulzbach
Telefonische Bestellung und Beratung: 07193 / 8699

PROFI FIBU

- schnelles Buchen
- Kontenplan
- Bilanz, G + V
- Statistik
- Kontenblätter
- Kein Kopierschutz
- Kontenplan
- **praktischer Maskeneditor**
- internationale Währungsbez.
- Formularbearbeitung
- Konten anlegen beim Buchen
- alle Mehrwertsteuersätze
- Dialog orientiertes Buchen
- 2000 Buchungen/Jahr
- und, und, und...

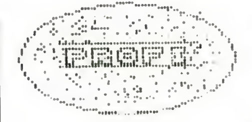
Ein Programm für Atari ST und Drucker

Preis: nur DM 298,–

DEMO inkl. Handbuch: DM 50,–
(wird bei Bestellung einberechnet!)

Auslandspreise auf ANFRAGE

Lieferung per NN + DM 8,–
bei Vorkasse, V.-Scheck frei!



PROFISOFT
Postfach 56
A-6027 Innsbruck

Das bewährte PC-Programm endlich auch für den ST!

WIR MACHEN UNMÖGLICHES MÖGLICH – ATARI ST –

'ST-Computer' – Public-Domain-Software zum fast unvorstellbaren Preis von **ab DM 2,–** pro PD-Disk-Nr. – gepackt zu Paketen von immer 4 kompletten Disk-Nrs. auf einer 2 S. – 3 5"-Diskette!!! (Disk-Nr. 1–4/5–8/9–12/.../125–128)

Preisstaffel:
Bei Abnahme von bis zu 4 Paketen – DM 10,– pro Paket; ab 5 Paketen kostet jedes Paket nur noch DM 8,– !!!

WIR LIEFERN AUCH FÜR 1-SEITIGE LAUFWERKE: PD-Disk-Nr. – gepackt zu Paketen von immer 2 kompletten Disk-Nrs. auf einer 1-S. – 3,5"-Diskette!!! (Disk-Nr. 1+2/2+3/5+6/7+8/.../121+122/123+124)
Bei Abnahme von bis zu 5 Paketen DM 7,– pro Paket; ab 6 Paketen kostet jedes Paket nur noch DM 6,– !!!

Das besondere Angebot – THE BEST –
Spiele-Paket: die 20 besten SPIELE-Disks auf fünf 2-S.-Disks **nur DM 40,–**
Utility-Paket: die 20 besten UTILITY-Disks auf fünf s-S.-Disks **nur DM 40,–**
Anwender-Paket: die 40 BESTEN ANWENDER-Disks auf zehn 2S.-Disks **nur DM 80,–**

Porto und Verpackung: bei Scheck DM 2,50 / Per Nachn. DM 5,–
Programmliste und Probedisk (mit vielen Prgs.) **gegen DM 5,–**
(bitte geben Sie an, ob 1- oder 2-seitige Diskette!!)

FsKS LUDWIG · Kastanienallee 24 · D-7600 Offenburg
Hotline 18.30 bis 19.30 unter ☎ 07 81 / 5 83 45



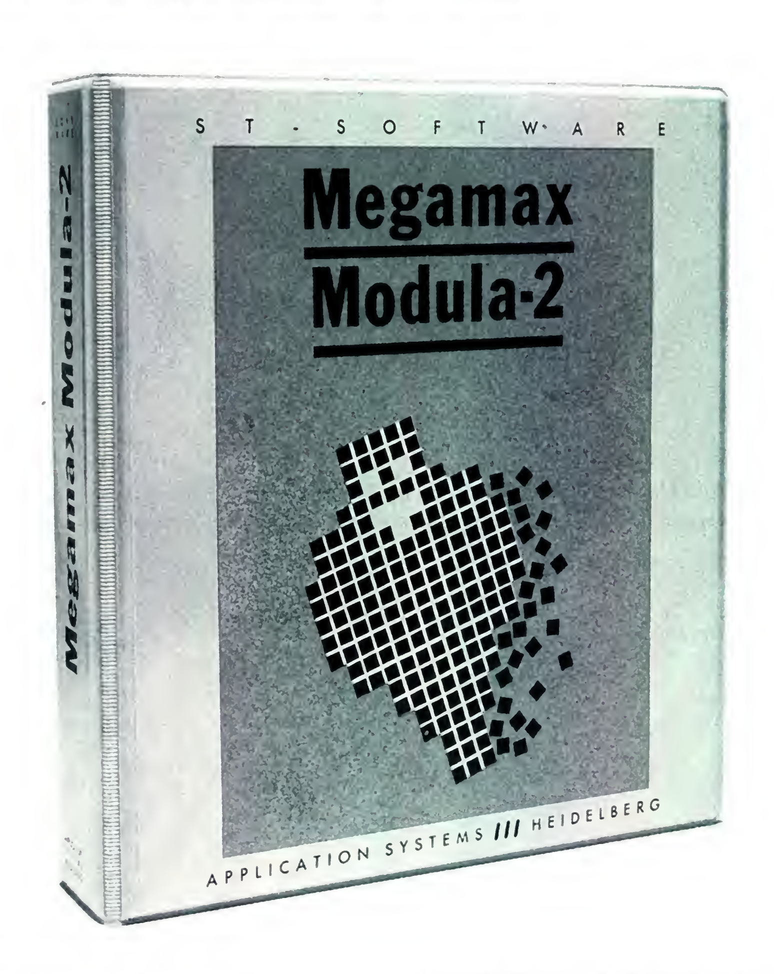
COMPUTERVERSAND WITTICH

Tulpenstr. 16 · 8423 Abensberg · Tel. (0 94 43) 4 53

ATARI 520 STM	498,–	SIGNUM	369,–
ATARI 1040 STF	998,–	NEC P2200	979,–
MEGA ST	auf Anfrage	STAR LC 10	559,–
ATARI SH 205	1098,–	DISK.-STATION SF 314	359,–
VORTEX HDPLUS 30	1398,–	NEC 1037-STATION	359,–
MONITOR SM 124	398,–	SCART-KABEL	38,–
ORIGINAL MAUS	98,–	ARCHIMEDES	3.598,–

NEU: MODERN SAMPLING **128,–**
Der Sampler der Soundsampling preiswert macht. Incl. Software und deutscher Anleitung auf Diskette

MEGAMAX MODULA-2



Nach längerem Warten haben Modula-2 Fans eine neue Alternative: Megamax Modula-2 ist erschienen. Das in der Bundesrepublik entwickelte Paket wartet mit einer ausgefeilten Arbeitsumgebung und umfangreichen Bibliotheken auf. Wir testeten, wie sich der Neuling in der inzwischen größer gewordenen Konkurrenz der Modula-Systeme schlägt.

Für DM 398,- erhält man von Application Systems Heidelberg vier Disketten, davon zwei doppelseitig. Änderungen gegenüber dem Handbuch sind in einer Info-Datei vermerkt.

Installation

Bei Diskettensystemen ist keine weitere Installation notwendig. Beim Booten wird automatisch eine RAM-Disk erzeugt und einige Dateien hineinkopiert. Das System wird dann von Diskette gestartet und greift auf die Dateien aus der RAM-Disk zu.
Zur Installation auf Festplatte müssen die entsprechenden Dateien kopiert werden. Dieser Vorgang nimmt ungefähr eine halbe Stunde in Anspruch, in der 1,5 MB an Dateien und Programmen bewegt werden.
Bei der Installation kann die Ordnerstruktur beliebig gewählt werden. Mit der Datei SHELL.INF können Suchpfade für alle Bestandteile frei festgelegt werden. Einen Ausschnitt aus dem File sehen Sie in Bild 1 bei der Bearbeitung im Editor.

Die Umgebung

Die über 200 KByte große Shell von Megamax Modula-2 (MM2) verhält sich auf den ersten Blick wie der Desktop mit einigen zusätzlichen Funktions-Icons (Bild 2). Laufwerke können durch Anklicken der Laufwerkssymbole geöffnet werden. In den Dateifenstern können alle Desktop-Operationen wie Kopieren, Umbenennen oder Info-Anzeige durchgeführt werden.
Der Umgang mit dem Modula-System ist sehr objektorientiert: Zum Edieren einer Source wird die Datei auf das "Edieren"-Icon geschoben und damit der Editor gestartet. Analog läuft der Aufruf des Compilers oder das Ausführen eines Programms ab. Wer intensiv Programme entwickelt, kann auf die Benutzung des GEM-Desktop völlig verzichten, da die Shell dessen Funktionen zur Dateipflege beinhaltet.
Da beim Programmieren die Mausbenutzung oftmals hinderlich ist, kann die Shell auch über die Tastatur benutzt werden. Anstelle des Anklickens auf Funktionssymbole treten Control-Tasten.
Aber die Umgebung ist mehr als eine reine Unterstützung beim Turn-Around zwischen Editor und Compiler. Megamax Modula-2 benutzt das Load-Time-Linking Konzept, das es

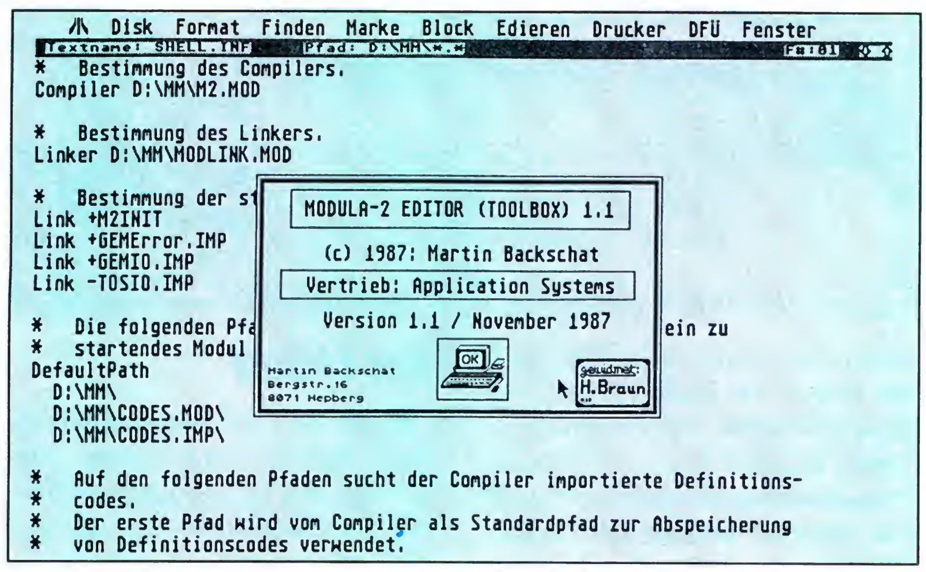


Bild 1: Die Konfigurationsdatei im Editor

in frühesten Modula-Zeiten schon einmal auf dem Apple II gab. Das Zusammenbinden der verwendeten Module geschieht erst beim Start des Hauptmoduls - im Gegensatz zum herkömmlichen Vorgehen, bei dem ein Linker ein komplettes Programm zur Ausführung erzeugt.

Beim Start eines Moduls erkennt die Umgebung, und das ist hier die Shell, welche Module importiert werden und führt den Bindevorgang beim Laden aus. Wenn drei Programme das InOut-Modul importieren, belegt es daher nur einmal Platz auf dem Massenspeicher. Der Linkerlauf muß dafür bei jedem Programmstart ausgeführt werden, ist allerdings sehr schnell, da ja nur im RAM gearbeitet werden muß.

Die Shell ist in der Lage, Module oder Programme resident im RAM bereitzuhalten, so daß Diskettenzugriffe entfallen können. Da für das Load-Time-Linking die Shell erforderlich ist, gibt es auch einen konventionellen Linker, der zum Abschluß der Programmentwicklung ein stand-alone Programm erzeugt.

Die komplette Shell liegt im Modula-Quelltext bei, so daß individuelle Änderungen kein Problem sind. Ein erfreulicher Service von Application Systems, der einem auch Einblicke in interne Vorgänge des Systems erlaubt.

Der Editor

Application Systems legt als Editor eine Version des auch einzeln erhältlichen Toolbox-Editors bei. Die Einbindung eines anderen Editors - die Anleitung schlägt hier Tempus vor - ist ohne weiteres über das genannte SHELL.INF möglich.

Der Editor ist nicht speziell auf Modula zugeschnitten. Er bietet alle Funktionen, die für das Programmeschreiben notwendig sind, wobei er erheblich schneller als der TDI-Modula Editor ist. Die verschiedenen Menüs sehen Sie in Bild 3. Der Großteil der Funktionen ist erfreulicherweise auch über Tastatur aufrufbar.

Neben den Edierfunktionen finden sich einige zusätzliche Features, so z.B. zehn verschiedene Textfenster und eine Terminalemulation. Der Toolbox-Editor vervollständigt das System und ist für eine komfortable Programmentwicklung angemessen.

Der Compiler

Der in Assembler geschriebene Compiler arbeitet mit einer erstaunlichen Geschwindigkeit. Im Vergleich mit TDI ist er ungefähr doppelt so schnell (Bild 4a).

Es handelt sich um einen Ein-Pass-Übersetzer nach dem neuesten Modula-Standard. Der REAL-Typ ist mit 64 Bit implementiert, was eine Genauigkeit von 13,5 Stellen ermöglicht. Dafür ist LONGREAL entfallen, was bei der Übernahme von Modula-Programmen unpraktisch ist.

Bei der Übersetzung können verschiedene Compileroptionen zur Überprüfung von Bereichsgrenzen etc. gesetzt werden. Das Einfügen von Sourcetext aus anderen Dateien ist über ein Include möglich. Mit $P+ kann ein ausführliches Protokoll erstellt werden, das sogar eine Angabe über die Compiliergeschwindigkeit enthält.

Bei den schon von der Besprechung des Jefferson Modula-Compiler bekannten Benchmarks schnitt Megamax Modula schlecht ab (Bild 4b). Alle Operationen, bei denen Ausdrücke ausgewertet werden, laufen langsamer als bei TDI ab. Grund dafür ist die völlig stackorientierte Berechnung von Expressions.

Der Code wird nicht weiter optimiert,

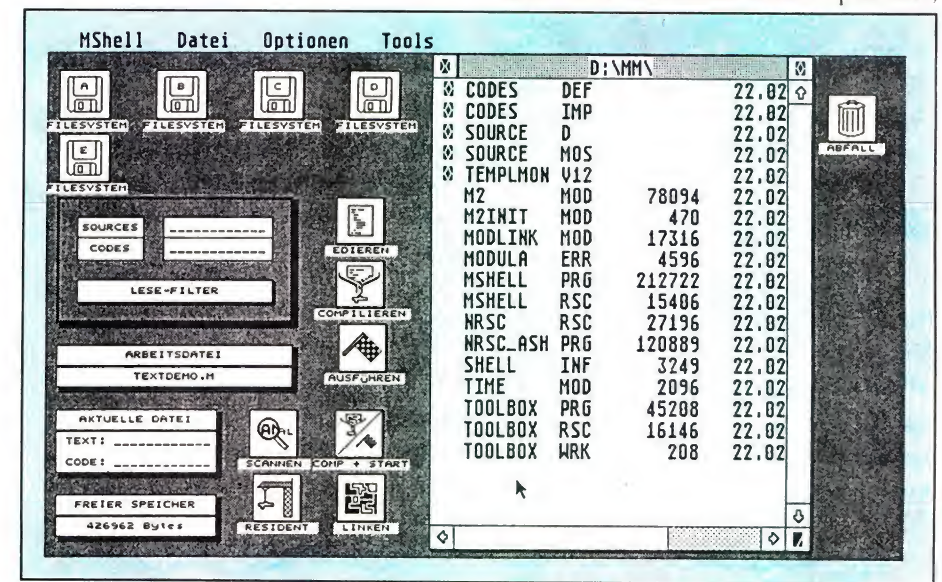


Bild 2: Die Shell

```
Disk
  Text laden            F10
  Text einfügen         E^e
  Text anhängen         E^a
  Text sichern          SF10
  Text sichern als      E^s
  Text drucken          E^d
  Text löschen          E^l
  Textname ändern       E^n
  ---
  Text-Infos            HELP
  Text verkl. zeigen
  Inhaltsverzeichnis
  ---
  Arbeit laden          E^r
  Arbeit sichern        E^o
  ---
  Compilieren           E^c
  Ausführen             E^x
  Programm beenden      E^q

Format
  Schriftart groß       CF1
  Schriftart klein      CF2
  ---
  TAB Spalte 8
  TAB Spalte setzen     CF9
  ---
  Text einfügen         F1
  Text überschr.        F2
  ---
  Automat. Einrücken
  Einrückung setzen     SF9
  ---
  langsamer Cursor      SESC
  schneller Cursor      CESC
  ---
  Drucker-Anpassung
  RS232-Konfiguration
  ---
  Funkt.-Tasten Menü

Finden
  Textanfang            SF1
  Textende              SF2
  Block-Anfang          SF5
  Block-Ende            SF6
  Goto Zeile ...        F4
  Goto Seite ...        E^g
  ---
  Bildschirmanfang      HOME
  Bildschirmende        SHOME
  ---
  Eine Seite zurück     S
  Eine Seite vor        S
  ---
  Akt. Wort suchen      CF3
  ---
  Ausdruck finden       SF3
  Weiterfinden          F3
  Ausdruck ersetzen     SF4
  Weiterersetzen        CF4

Marke
  Marke #1 setzen       E^1
  Marke #2 setzen       E^2
  Marke #3 setzen       E^3
  Marke #4 setzen       E^4
  ---
  Marke #1 anspringen   A-1
  Marke #2 anspringen   A-2
  Marke #3 anspringen   A-3
  Marke #4 anspringen   A-4

Block
  Start definieren      F5
  Ende definieren       F6
  Text als Block        E^T
  Block abschalten      E^A
  ---
  Block löschen         E^L
  Block einrahmen       E^R
  Block einrücken       E^E
  Block kopieren        E^K
  ---
  Block Spaces -> Tabs
  Block Tabs -> Spaces
  Block-Umlautwandlung
  ---
  Block drucken         E^D
  Block sichern         E^S

Edieren
  Zeile entfernen       SF7
  Zeile einfügen        F7
  Seite entfernen       SF8
  Seite einfügen        F8
  ... Zeilen entfernen
  ... Zeilen einfügen
  ---
  Bis Cursor löschen    SDEL
  Ab Cursor löschen     SINS
  Akt. Wort löschen     CUNDO
  Zeile löschen         SUNDO
  ---
  Zeile zentrieren      F9
  ---
  TAB-Zeichen setzen    STAB

Drucker
  Bildschirm-Hardcopy
  ---
  Ausdruck mit Steuerkodes

DFÜ
  Online...             CF10
  Offline               UNDO
  ---
  Text empfangen
  Text anhängen
  ---
  gr. Schrift           F1
  kl. Schrift           F2
  ---
  Text senden           F3
  File senden           F4
  ---
  DÜ-Parameter          F5
  F-Tasten-Menü         F6

Fenster
  Anlegen...
  Löschen...
  --- ESC+Fx ---
  Fenster 1
  Fenster 2
  Fenster 3
  Fenster 4
  Fenster 5
  Fenster 6
  Fenster 7
  Fenster 8
  Fenster 9
  Fenster 10
```

Bild 3: Die Editor-Funktionen

benutzt keine Register für die Speicherung von Zwischenergebnissen und läßt konstante Ausdrücke zur Laufzeit berechnen. An dieser Stelle hat das System eine ordentliche Verbesserung nötig.

Hervorragend schnell hingegen sind die REAL-Berechnungen implementiert. Megamax Modula hat im Vergleich mit den gleichlangen LONGREAL von TDI um ein Viertel der Rechenzeit die Nase vorne.

Die Bibliotheken sind, da teilweise in Assembler optimiert ebenfalls überaus flink. Die getesteten String-Operationen sind doppelt so schnell wie bei den anderen verfügbaren Modula-Systemen.

Trifft der Compiler auf einen Fehler, wird die Übersetzung abgebrochen und man hat die Möglichkeit, sofort den Editor zu starten. Dabei wird der Cursor auf die entsprechende Stelle gesetzt und die Fehlermeldung angezeigt. Wer Turbo-Pascal kennt, weiß eine solche Integration zu schätzen.

Allerdings wäre es wünschenswert, wenn der Compiler nach einem Fehler wieder aufsetzen könnte, um alle Fehler im Sourcecode zu ermitteln. Auch wenn der Wechsel Compiler-Editor-Compiler sehr schnell und ohne den Umweg über die Shell möglich ist, würde ein solches Vorgehen in vielen Fällen zeitsparend sein.

Der Assembler

Megamax Modula-2 integriert einen vollständigen 68000-Assembler. Dazu wurde als Statement ein ASSEMBLER...END-Block in die Syntax aufgenommen. Innerhalb dieses Blocks kann ein Maschinenprogramm mit Motorola-Mnemonics notiert werden.

In den Assemblerstatements lassen sich die im Modula-Programm definierten Bezeichner verwenden; auch der Aufruf von Modula-Prozeduren ist kein Problem. Das Handbuch erläutert detailliert Registerbenutzung, Parameterübergaben und den Zugriff auf Modula-Variablen und -Datenstrukturen wie Arrays oder Records. Gerade für die Geschwindigkeitsoptimierung ist der Assembler außerordentlich hilfreich. Da der volle 68000-Befehlssatz unterstützt wird, kann das Modula-System auch als Assembler benutzt werden. Auch wenn Puristen wahrscheinlich wegen der Prozessorabhängigkeit aufschreien werden, ist diese Erweiterung ideal für schnelle Programme. Die Routinen werden hochsprachlich formuliert und ausgetestet und dann in zeitkritischen Teilen mit Assembler ausprogrammiert. Überhaupt erschließt sich hiermit für Modula-Programmierer endlich die Assemblerwelt. Bisher war eine Benutzung der bekannten Assembler-Systeme aufgrund der völlig unterschiedlichen Linkformate nicht möglich.

Der Linker

Da für das Load-Time-Linking die Shell benötigt wird, müssen Programme, die weitergegeben werden, mit einem herkömmlichen Linker zu .PRG-Dateien gemacht werden.

Der Linkvorgang (Bild 5) dauert etwa so lange wie beim TDI-Linker. Allerdings werden die Programme nicht optimiert - Routinen aus den externen Modulen werden auch dann eingebunden, wenn sie nicht importiert werden. Das Ergebnis sind Programme, die merkbar länger werden, als beim TDI-System.

Beim TDI-Linker, der eine Option zum Ein- und Ausschalten der Optimierung besitzt, hat sich gezeigt, daß unoptimierte Programme zwischen 20 und 40 Prozent länger als nötig sind.

Der Linker erfüllt seine Aufgabe zufriedenstellend; eine Optimierung wäre wünschenswert und soll wohl auch in Arbeit sein.

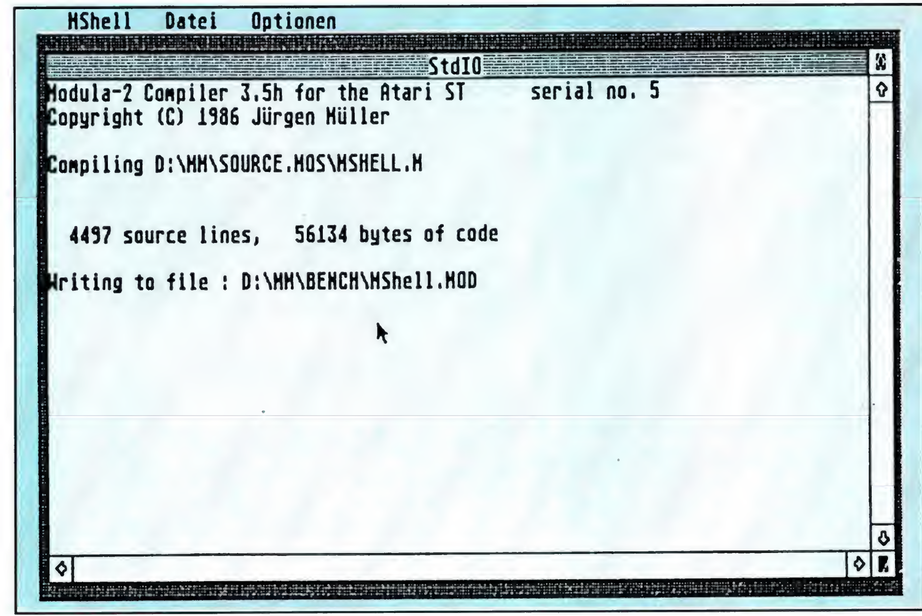


Bild 4a: Der Compiler bei der Arbeit ...

Modula-2 Benchmarks							
Nr.	TDI Modula-2		JS Modula-2		MM Modula-2		testet ...
	Min.	Ticks	Min.	Ticks	Min.	Ticks	
1	0:07	1334	0:07	1489	0:07	1480	Prozeduraufruf
2	1:42	20339	1:33	18548	2:59	35753	Addition
3	1:21	16277	1:18	15518	1:58	23506	Increment
4	1:47	21418	1:38	19634	2:59	35839	Additionsoptimierung
5	1:27	17311	1:23	16550	2:08	25554	Increment als Vergleich
6	2:09	25835	1:57	23480	3:48	45580	INTEGER-Addition
7	2:09	25833	1:57	23471	3:48	45581	CARDINAL-Addition
8	1:11	14228	1:18	15543	1:42	20437	FOR-Schleife
9	1:21	16278	1:02	12437	1:42	20398	REPEAT-Schleife
10	1:21	16191	1:18	1552	1:57	23487	WHILE-Schleife
11	1:04	12743	0:54	10791	1:15	15057	INTEGER-Parameter
12	1:04	12739	0:54	10791	1:17	15472	INTEGER VAR-Parameter
13	1:06	13130	0:59	11821	2:19	27807	RECORD-Parameter
14	0:34	6755	0:30	5977	0:41	8167	RECORD VAR-Parameter
15	0:49	9837	0:49	9780	1:33	18556	Konstanten-Optimierung
16	0:51	10249	0:51	10201	1:33	18552	Konstanten-Optimierung
17	1:28	17655	1:26	17171	2:06	25117	Expression-Optimierung
18	1:42	16822	1:22	16354	1:59	23884	Expression-Optimierung
19	0:37	7431	0:36	7102	0:55	10933	Zwischenergebnis-Optimierung
20	0:37	7435	0:35	7099	0:55	10931	Zwischenergebnis-Optimierung
21	0:09	1831	0:11	2154	0:13	2608	IF-Statement
22	0:13	2645	0:13	2688	0:16	3226	IF durch CASE ausgedrückt
23	0:38	7689	0:33	6667	0:41	8188	CASE-Statement
24	0:40	7960	0:39	7804	1:03	12630	CASE durch IF ausgedrückt
25	0:47	9441	1:03	12636	—	—	REAL-Arithmetik
26	2:05	24924	—	—	1:32	18495	LONGREAL-Arithmetik
27	1:52	22380	5:42	68358	—	—	REAL-Library
27a	5:39	67797	—	—	2:35	30990	LONGREAL-Library
28	1:21	16264	1:21	16135	0:40	8016	String-Library
29	2:10	26075	2:07	25437	2:13	26618	ARRAY-Zugriffe
30	0:09	1896	0:10	2056	0:17	3398	RECORD-Zugriffe
Alle Zeiten mit time-Kommando von Gulâm gemessen							

Bild 4a: Die Benchmarks

Die Bibliotheken

Die über 70 Bibliotheksmodule gehen weit über die üblicherweise vorhandenen GEM-Module hinaus. Bei den AES- und VDI-Libraries werden für alle Datenstrukturen eigene Record-Typen vordefiniert, die für eine übersichtliche GEM-Programmierung sorgen. So werden z.B. alle Messagearten eines EventMessage-Calls in einem varianten Record dargestellt, so daß man beispielsweise bei einer Redraw-Aufforderung das neu zu zeichnende Rechteck in einem Rectangle-Typ erhält. Auch wenn sehr viele Bezeichner zu merken sind, werden GEM-Programme dadurch endlich lesbarer.

Was man beim TDI-Compiler als Toolkit dazukaufen muß, ist hier standardmäßig vorhanden. Es stehen Module für die Event-Verwaltung und Textfenster oder die Manipulation von Objektbäumen zur Verfügung. Ein Beispiel für den Komfort der Routinen ist die Anmeldung einer GEM-Applikation, die automatisch eine Workstation öffnet und alle notwendigen Handles zurückgibt.

Für systemnahe Programme gibt es Prozeduren, die in sauberem Hochsprachenstil z.B. das Setzen der Exception-Routinen oder den Zugriff auf Systemvariablen ermöglichen.

Die Modula-Standard-Module liegen in erweiterter Form vor. Das Filesystem kann mit mehreren Suchpfaden arbeiten, die Konvertierroutinen erkennen automatisch Zahlen, die mit "$" oder "%" als hexadezimal oder binär ausgezeichnet sind. Das InOut-Modul eröffnet für Textausgaben ein GEM-Textfenster, somit gehören TOS-Programme wohl der Vergangenheit an. Dafür braucht der Programmierer keinen Finger zu rühren; alle Routinen sind darauf vorbereitet. GEMDOS wird mit vielen Zusatzfunktionen unterstützt. So sind z.B. für Directory-Manipulationen eigene Prozeduren vorhanden, die das Ändern von Dateiattributen, einfaches Ermitteln und Setzen des aktuellen Ordners oder auch das Aufteilen eines Filebezeichners in Namen und Extension übernehmen.

Zur einfacheren Übernahme von TDI-Programmen sind auch Module für GEMDOS, BIOS und XBIOS vorhanden, die identische Bezeichner verwenden. Allerdings wird ob der besseren Megamax-Unterstützung für GEMDOS von deren Verwendung abgeraten.

Viele kleinere Module enthalten die Routinen, die man immer schon schreiben wollte und fast immer benötigt. So z.B. das Konvertieren eines Datums im GEMDOS-Format in einen String wie "9. März 1988".

Auch weitere Module zur Bearbeitung von komplexeren Datenstruktu-

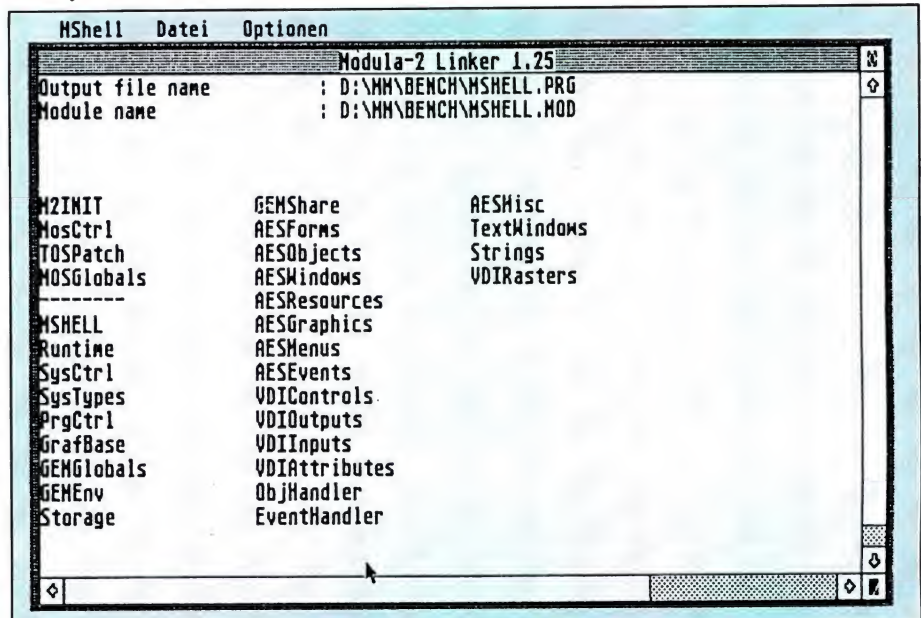


Bild 5: ... und der Linker am Schuften

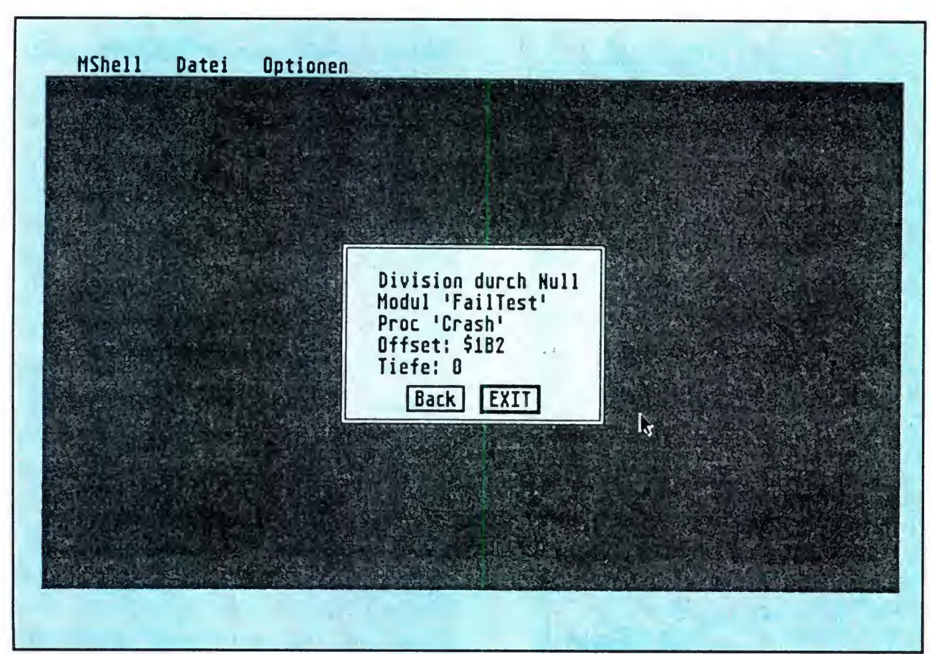


Bild 6: Ein Laufzeitfehler ...

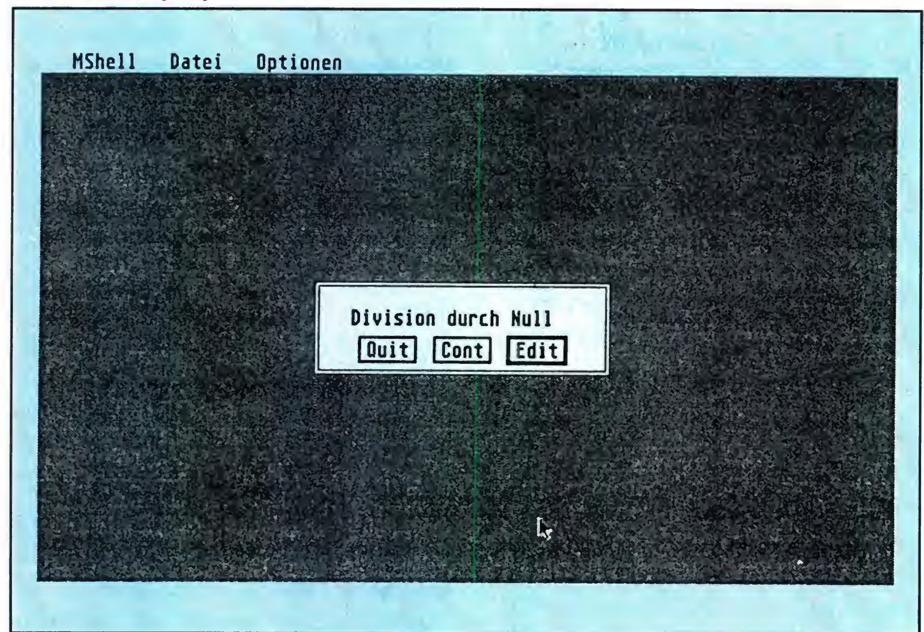


Bild 7: ... seine Folgen bei Megamax Modula-2 ...

```
 Disk  Format  Finden  Marke  Block  Edieren  Drucker  DFÜ  Fenster
Division durch Null
MODULE FailTest;

VAR i,j:INTEGER;

MODULE FailingModule;

EXPORT Crash;

VAR x,y:INTEGER;

PROCEDURE Crash;
BEGIN
  x:= x DIV y;
END Crash;

BEGIN
  x:=1;
  y:=0;
END FailingModule;

BEGIN
  Crash;
END FailTest.
```

Bild 8: ... und seine Anzeige im Editor

ren wie Listen und Bäume werden mitgeliefert. Der Umfang, die Konzeption und die Funktionalität der Bibliotheken ist beeindruckend und übertrifft das TDI-System bei weitem. Ein Modula-System lebt auch von den vorhandenen Libraries - Megamax Modula zeigt, wie ein Entwicklungssystem dem Programmierer Arbeit abnehmen sollte.

Entwanzen

Das Debugging ist bis jetzt bei keinem Entwicklungspaket befriedigend gelöst worden. Megamax Modula-2 glänzt hier durch ausführliche Fehlermeldungen, die Anzeige der Fehlerstelle im Quelltext und eine Einzelschrittausführung mit Source-Anzeige.

Beim Auftreten eines Laufzeitfehlers werden in einer Alertbox umfangreiche Informationen über die Fehlerstelle gegeben (Bild 6). Angezeigt werden die Fehlerart, verursachendes Modul und die Prozedur, in der der Fehler entstand. Mit den Buttons "Back" und "Frwd" hat man dann die Möglichkeit, die Aufrufkette zu inspizieren.

Bei Abbruch durch "Exit" erscheint eine weiter Auswahl (Bild 7), mit der man entweder zur Shell zurückkehren, das Programm unter Ignorierung des Fehlers weiterführen oder den Scanner starten kann.

Der Scanner verbindet die Fehler-Informationen mit dem Quelltext und startet dann den Editor. Dieses Vorgehen erinnert stark an Turbo-Pascal und führt zu einem Ergebnis wie in Bild 8. Der Editor zeigt die Stelle im Sourcetext samt der Fehlermeldung an. Schon damit wird das Debugging zum Kinderspiel und erspart viele Überlegungen.

Falls die Fehlerursache nicht so offensichtlich ist, wie in dem abgebildeten Beispiel, bietet Megamax Modula-2 ein weiteres Bonbon: Durch Importieren des Debug-Moduls und Setzen der Compileroption $D+ lassen sich die Programme im Einzelschrittmodus ausführen (Bild 9). Dabei wird das Debug-Modul durch zusätzlich in den Code eingefügte Traps nach jedem Programmschritt

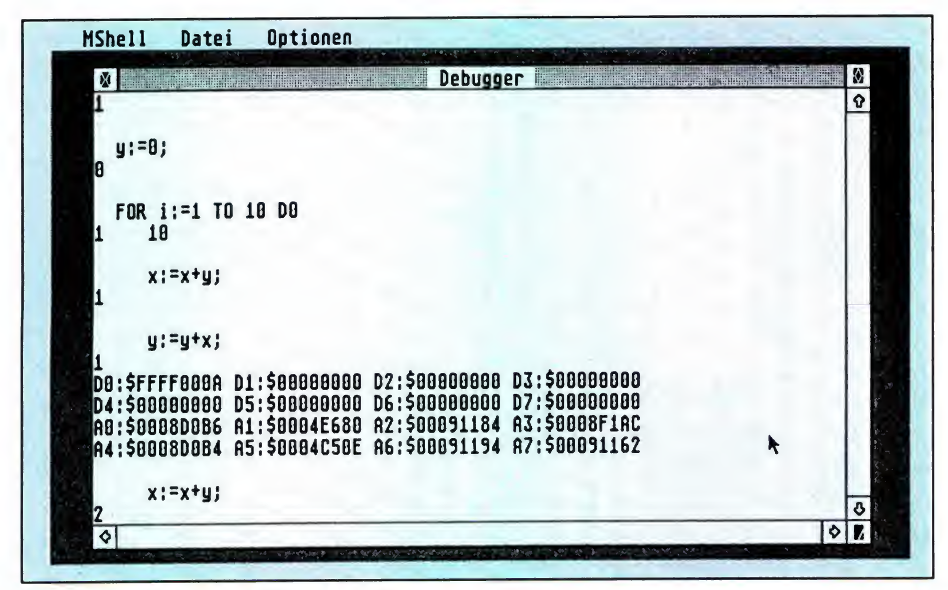


Bild 9: Einzelschritte in Hochsprache

aufgerufen. In einem Fenster werden dabei die jeweilige Quelltext-Zeile und die Ergebnisse der gerade berechneten Ausdrücke angezeigt.
Daraufhin mögliche Kommandos zeigen z.B. die aktuellen Registerinhalte oder die Codeadresse der Instruktionen an. Ausgaben können dezimal oder hexadezimal notiert werden. Durch verschiedene Steuervariablen des Debug-Moduls lassen sich die Untersuchungen in einem Programm auf einen bestimmten Block oder Programmteile beschränkten, so daß potentielle Fehlerquellen ausführlich überwacht werden können.
Der Debugging-Teil des Systems ist für ein Compilersystem geradezu ideal. Die Fehlersuche läßt sich adäquat abstufen und sollte jedes auch noch so unerklärliches Problem lösen. Megamax Modula-2 übertrifft hier alle anderen Entwicklungspakete um Längen und bietet dem Programmierer eine bisher unerreichte Unterstützung.

Was noch ?

Inzwischen haben die Vertreiber von Entwicklungssystemen eingesehen, daß zu einem Compiler für den Atari ST ein Resource-Construction-Set gehört. So enthält Megamax Modula-2 im Gegensatz zum TDI-Modula das schon von Lattice-C und anderen Entwicklungssystemen bekannte NRSC (Bild 10).
Das Programm ermöglicht auf sehr komfortable Weise die Erstellung von Resourcen, ohne die GEM-Programme nicht möglich sind. Die im Original von Kuma stammende Applikation liegt in der Version 1.1 als deutschsprachige Anpassung vor, die auch mit dem Blitter und dem Mega ST arbeitet.
Einige der Bibliotheksmodule sind in Source-Text beigelegt. Für Assemblerprogrammierer befindet sich der aus der Public-Domain bekannte Maschinen-Monitor TEMPLMON in der Version 1.12. Kein Wunder, denn Thomas Tempelmann gehört neben Jürgen Müller und Manuel Chakravarty zu den Entwicklern.
Diverse Beispielprogramme demonstrieren die Fähigkeiten des Systems und geben Hilfestellung bei systemnahen Tricks.

Das Handbuch

Das Handbuch kommt in einem Ringbuch - leider keine Spiralbindung - und ist vorbildlich aufgemacht. Im Textteil werden auf über 130 Seiten die Benutzung des Systems, der Compiler, der Assembler und die Bibliothekskonzepte ausführlich beschrieben. Der sehr detailliert geschriebene Text bietet sowohl dem Einsteiger als auch dem Experten alle Informationen über das System.
Die harten Fakten, wie Fehlermeldungen oder der Aufbau der Codemodule, finden sich in dem umfangreichen Anhang. Dabei nehmen die Listings der Definitionsmodule der Libraries den größten Teil ein. Die hervorragenden Kommentierungen zu dem Routinen klären alle Fragen, die im Textteil eventuell noch offen geblieben sind. Beim Umfang der Bibliotheken wird der Anhang zu einer wahren Fundgrube.
Die letzten beiden Anhänge enthalten eine Kreuzreferenz aller in den Bibliotheken verwendeten Bezeichner und ein 19-seitiges Register zum Textteil. Das Handbuch wird dadurch zu einem exzellent zu benutzenden Nachschlagewerk.
Das Manual ist hervorragend gelungen. Wenn ein Handbuch ideal sein kann, dann dieses. Man würde sich wünschen, wenn andere Softwarehersteller sich hieran ein Vorbild nehmen würden.

Support

Von einem in der BRD entwickelten Produkt kann der Anwender zu Recht eine weiterreichende Unterstützung erwarten. Über eine Münchner Mailbox ist es inzwischen möglich, direkten Kontakt mit den Entwicklern aufzunehmen. Sinnvolle Änderungswünsche sollen gegen eine geringe Aufwandsentschädigung durchgeführt werden.
Die von Application Systems schon bei Signum! angewandte preiswerte Update-Politik soll auch bei Modula-2 weitergeführt werden. Der Distributor will bei Interesse den Vertrieb von nützlichen Utilities zum System - wie bei Megamax-C - übernehmen.

Fazit

Arbeitsumgebung der Shell ist sehr praktisch und konzeptuell neu. Sie ist weit besser, als der TDI-Desktop. Der Compiler arbeitet sehr flott und bietet schnelle und genaue Fließkommaoperationen. Der erzeugte Code ist etwas langsam und müßte besser optimiert werden.
Mit dem Assembler lassen sich auch zeitkritische Programme verwirklichen. Das Paket ist zudem als Hochsprache und als Übersetzer für Bitfanaktiker doppelt nutzbar.
Der Linker erfüllt seine Aufgabe; im

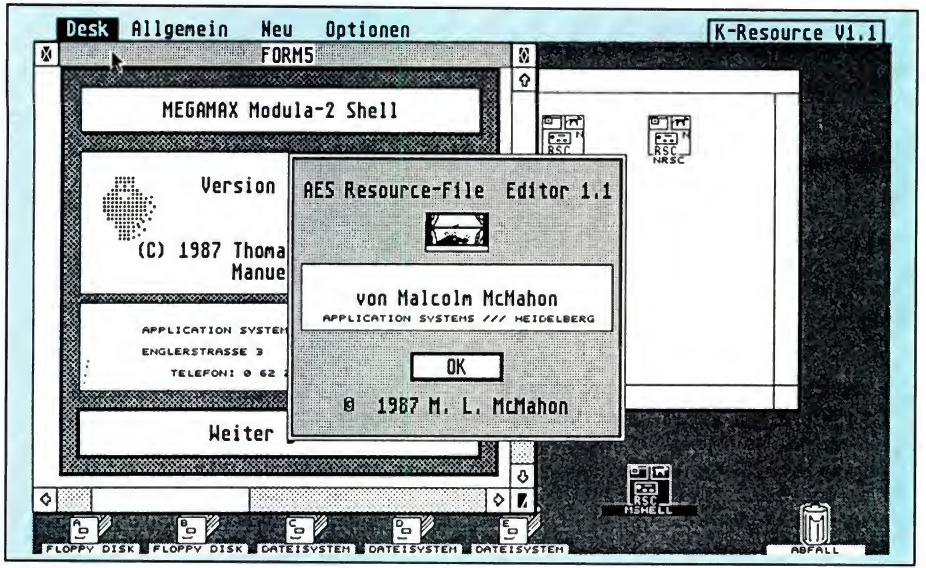


Bild 10: Resource-Entwurf mit NRSC

Hinblick auf Optimierung läßt er sich noch verbessern. Das Debugging-Konzept sucht seinesgleichen; es läßt sich für den Atari ST kaum noch verbessern.

Die mitgelieferten Bibliotheksmodule lassen kaum noch Wünsche offen, und auch wer viel programmiert wird selten zusätzliche Module schreiben müssen. Mit Editor und Resource-Construction-Set ausgerüstet ist das Paket für jede Programmieraufgabe optimal ausgestattet. Das Handbuch läßt keinen Wunsch offen und ergänzt das System nicht nur, sondern wertet es auf.

Das lange Warten auf Megamax Modula-2 - hauptsächlich durch die Programmierung der Shell und der Bibliotheken verursacht - hat sich gelohnt. Es handelt sich im Ganzen auch trotz der genannten Wermutstropfen Ausdruckauswertung und nichtoptimierender Linker um das beste Modula-2 System für den Atari ST. Der Support von Application-Systems läßt erwarten, daß auch diese letzten Flecke auf der ansonsten überaus weißen Weste beseitigt werden.

Robert Tolksdorf

Ein-Pass-Übersetzer für Modula-2

Modula-2 hat eine längere Entwicklung hinter sich. Vater der Sprache ist Niklaus Wirth, der auch schon Pascal entworfen hat. Die allererste Implementierung wurde auf der von Wirth konzipierten Workstation "Lilith" vorgenommen. Zu diesem Zeitpunkt war die Sprache wohldefiniert und speziell auf die Architektur dieses Rechners ausgelegt. In einer ersten Revision wurde kleine Änderungen an der Syntax vorgenommen, die vorerst als Standard-Modula galten. Der TDI-Compiler ist eine Implementierung dieser Version.

Inzwischen existiert eine dritte Sprachdefinition, ebenfalls von Wirth, in der wiederum kleine Änderungen enthalten sind, die allerdings nur kosmetischer Natur sind. Die wirkliche Änderung liegt in der Abstimmung der Sprache auf Ein-Pass-Compiler. Dabei geht der Übersetzer nur einmal über den Quelltext. Das hat allerdings zur Folge, daß jeder benutzte Bezeichner, sei es für Prozeduren, Typen oder Variablen vorher definiert werden muß (dies was in der "alten" Version nicht notwendig).

Das neue Schlüsselwort FORWARD erlaubt es, einen Bezeichner zu deklarieren, ohne z.B. die Prozedur schon auszuformulieren. Ein Name kann nur so auch dann verwendet werden, wenn er textuell nach der Verwendung deklariert ist.

Diese Änderung ist leider ein Schritt rückwärts und wohl mit Wirths Eigenheit zu begründen, schon bei der Sprachdefinition die konkrete Implementierung im Auge zu haben. Denn durch die Einführung der FORWARD-Deklarationen wird es möglich, Ein-Pass-Compiler zu schreiben. Sie sind logischerweise schneller als ihre Vorgänger mit mehreren Pässen, da ein Programmtext "in einem Rutsch" übersetzt wird.

Alle neueren Modula-Compiler halten sich an den neuesten Standard, so auch Megamax und Jefferson. TDI-Modula soll inzwischen auch in einer Ein-Pass-Version erhältlich sein.

ENDE

EXTENDED VT52-EMULATOR TEIL 2

In der letzen Folge ging es um die Installation des Emulators und die Problematik mit dem GEM-Vektor. Diesmal werde ich Ihnen u.a. einige der neuen bzw. erweiterten Escape-Sequenzen vorstellen.



Bevor ich mit der weiteren Besprechung des Programms fortfahre, möchte ich Ihre Aufmerksamkeit zunächst einmal auf die in Tabelle 1a und 1b dargestellten globalen Variablen TCB und CCB lenken. Da fast alle Funktionen auf diese Strukturen zugreifen, ist es für das Verständnis des Programms nicht ganz unwichtig, zu wissen, welche Informationen sie enthalten. Im allgemeinen wird auf beide Strukturen nur über Offsets zur Basisadresse zugegriffen. Weil sich das so herzerfrischend wissenschaftlich und also chinesisch anhört, hier ein Beispiel: Zu Beginn einer Funktion wird die Adresse des Cursor-Control-Blocks CCB mit dem Befehl "lea CCB,a4" in ein Adress-Register des MC68000 geladen. Möchte man nun die absolute Cursorposition (irgendeine Adresse im Video-RAM) eruieren, so kann man mit 6(a4) darauf zugreifen. Im Prinzip ist das nichts anderes als die Assembler-Variante des von Pascal und C her bekannten Zugriffs auf Records/ Structs und Unions, also z.B. abs_curs(:)=CCB.address. Sie sehen, daß das Verständnis der Maschinensprache manchmal ganz nützlich sein kann, wenn es darum geht, sich abstrakte und damit nicht immer leicht verständliche Sprachkonstrukte zu vergegenwärtigen, wobei dies insbesondere für die leidigen Pointer gilt! Ich nehme an, daß Sie dank der Kommentare nun keine Schwierigkeiten mehr haben werden, die beiden Kontrollvariablen zu verstehen. Aber noch eine weitere fundamentale Vorbetrachtung ist zum Programmverständnis notwendig, nämlich die Organisation der systemeigenen

Fonts

Was nützt ein mit allen gängigen Superlativen dekorierter Computer, wenn er sich nicht in der Lage sieht, sich im Klartext mit dem Benutzer zu unterhalten? Gewieft, wie die Hersteller nun 'mal sind, haben sie Mittel und Wege ersonnen, die uns allen bekannten Schriftzeichen zur Darstellung zu bringen. Dabei gehen allerdings sowohl Mittel wie Wege ihre eigenen ebensolchen. Hersteller X zieht es beispielsweise vor, den ganzen Zeichenkrempel der Hardware zu überlassen; entsprechende Chips hören auf den Namen Character Generators. Der Vorteil hierbei liegt sicherlich in der Einfachheit der Bedienung und der Darstellungsgeschwindigkeit. Man übergibt dem Chip schlicht die Nummer des Zeichens, das darzustellen man sich wünscht - und läßt die Transistoren pfriemeln. Die Nachteile sind aber auch nicht ganz ohne: abgesehen davon, daß selbst Chips "Made in Taiwan" nicht für umsonst zu haben sind, erlauben nur einige wenige den Gebrauch von Textattributen (hell, blinkend, kursiv...), und dann haben sie auch ihren Preis. Ergo gehen andere Hersteller den Weg über die Software. Die ist zwar auch nicht gerade billig, aber wenn sie erst einmal erstellt ist, kann man sie beliebig oft und quasi gratis ohne Qualitätsverlust vervielfältigen (was zugegebenermaßen nicht jedem gefällt - die verschiedensten Kopierschutzmechanismen sprechen da eine deutliche Sprache...). Wie funktioniert nun aber die softwaremäßige Zeichendarstellung? Die Voraussetzung hierzu ist auf jeden Fall, daß der Bildschirm zur Kategorie "bitmapped" gehört.

Bitmapping?

Weil man sich darunter wieder schrecklich viel vorstellen kann, sei's kurz erläutert. Der SM124 bringt am ST 640 mal 400 Pixel (Abk. für: picture elements) zur Darstellung. Jeder dieser Bildpunkte wird durch ein Bit im Hauptspeicher des Rechners repräsentiert, d.h., der Bildschirm ist nichts anderes als ein durchgehender RAM-Bereich einer durch die Auflösung bestimmten Länge. Beim Atari sind dies 640*400=256000 Bits durch 8 macht 32000 Bytes. Möchte man nun einen Punkt setzen, wird im mit diesem Punkt korrespondierenden Byte einfach das entsprechende Bit auf "1" gesetzt (raten Sie mal, wie das Löschen funktioniert!). Ein besonderer Chip ist darauf spezialisiert, das Video-RAM ständig abzutasten und die entsprechenden Signale zu erzeugen, die der Bildschirm dann letztendlich darstellen kann. So funktioniert's bei monochromatischer Darstellung; kommt Farbe ins Spiel, wird's schon kniffliger. Will man beispielsweise vier Farben darstellen, so benötigt jeder Bildpunkt schon 2 Bit an Information. Die entsprechende Bitkombination muß dann ebenfalls von der Hardware interpretiert werden. Soll der Bildspeicher weiterhin seine Größe von 32000 Bytes behalten, läßt es sich nicht vermeiden, daß sich die Anzahl der darstellbaren Pixel halbiert! Das ist auch der Grund, warum man in der mittleren Auflösung zwar vier Farben, aber nur noch 640 mal 200 Bildpunkte darstellen kann. Übrigens nennt man die Gesamtheit eines Bits über alle Pixel eine "bitplane" (bei Monochrombetrieb gibt's nur eine, bei vier Farben zwei, bei sechzehn Farben vier, bei... na, Sie wissen schon: Logarithmus dualis und so).

Zeichen auf dem Monitor

Nach diesem Ausflug in die Welt der "bitmapped devices" können Sie sich sicher leicht vorstellen, wie man nun Zeichen auf den Monitor bringt. Jedes Zeichen wird durch eine Kombination aus gesetzten oder eben nicht gesetzten Bits repräsentiert, und zwar in einem Feld von soundsoviel mal soundsoviel Punkten. Im einfacheren Fall ist die Größe des Rasters für alle darstellbaren Zeichen konstant. Im Atari gibt's gleich drei System-Zeichensätze (Fonts), bei denen dies so gelöst ist (6x6, 8x8 und 8x16). Daß es aber auch anders geht, wissen Kenner von Apple's Mac zu berichten (hat da jemand Aladin gesagt?). Bei diesen Rechnern hat man nicht nur beliebig viele Zeichensätze zur Verfügung, sie sind auch alle für Proportionalschrift ausgelegt, d.h. ein 'i' benötigt weniger Platz als ein 'M'. Im Endeffekt sieht das dann zwar ein wenig ästhetischer aus, ist aber wesentlich aufwendiger. Stellen Sie sich vor, Sie müssen ein Zeichen löschen: abgesehen davon, daß es nicht ganz einfach ist, den Cursor genau auf das Zeichen zu setzen - wieviele Pixel sind's denn nun, die auszuradieren sind? Ich nehme an, daß Franze "SIGNUM!" Schmerbeck (Tagchen auch) hier ein paar Klagelieder anzustimmen weiß... Bleiben wir also lieber bei der Darstellung äquidistanter Zeichen.



Umschalten vom GEM-Fonts

Fonts im ROM

Problem: wie legt man die Font-Daten im ROM ab? Exemplarisch sei's am 8x16-Font erklärt (bei den anderen Fonts ist es genauso). Man zerlegt hierzu die Zeichen in sogenannte "scanlines", wovon es logischerweise 16 gibt. Ab der Startadresse der Fontdaten stehen dann hintereinander: die oberste Zeile von chr(0), gefolgt von der obersten Zeile von chr(1), gefolgt... und so geht das denn bis 255. Dann kommt die zweitoberste Zeile von chr(0) und so weiter. Glücklicherweise ist nun eine Scanline genau acht Bit breit, so daß man sehr simpel auf ein Zeichen zugreifen kann: die oberste Zeile findet man an der Adresse FontStartAdresse + ASCII-Nummer(Zeichen). Die zweite befindet sich 256 Bytes weiter, die dritte... Nun weiß man aber von GEM/VDI-Programmen, daß auch ein ST zu Textattributen bis hin zur Proportionalschrift befähigt ist. Zu diesem Behufe ist wie üblich wieder Verwaltung notwendig, welche im "Fontheader" residiert. Dort findet man z.B., welche Zeichen überhaupt im Font gespeichert sind, wo die Startadresse des Fonts liegt, wie er heißt und einen Zeiger auf die sogenannte "Offset-Tabelle".

In dieser ist nun gespeichert, wieviele Bits jedes Zeichen breit ist. Somit müssen nur die Scanlines des ersten darstellbaren Zeichens auf einer Wortgrenze (gerade Adresse) beginnen; die restlichen Daten können über die Offset-Tabelle gefunden werden, was jedoch recht zeitaufwendig und rechenintensiv ist. Für den VT52-Emulator wie auch für die VDI-Escapes ist das aber alles vollkommen wurscht: sie erwarten, daß jedes Zeichen exakt ein Byte breit ist und damit basta! Ich habe die Existenz der Offset-Tabelle lediglich der Vollständigkeit halber erwähnt; sie ist, wie gesagt, für den Teil der Textausgabe, um den es hier geht, nicht relevant.

Was ich Ihnen nun lang und breit erklärt habe, ist die in Prosa gefaßte Prozedur PUT, die Sie am Anfang des Listings wiederfinden können. Abgesehen davon, daß sie ein Zeichen auf den Bildschirm bringt, sorgt sie auch dafür, daß - je nach eingestellten Attributen - das Zeichen unterstrichen, halbhell (disabled) oder invertiert dargestellt wird, wobei diese Attribute auch kominiert werden können. Wie man diese Attribute wählen kann, erfahren Sie im folgenden Kapitel.

Massenweise Esc-Sequenzen!

xVT52 erweitert die Anzahl der Steuerzeichen auf mehr als das Doppelte. Das bedeutet zwar, daß Programme, die von den neuen Möglichkeiten Gebrauch machen, nicht mehr ohne den erweiterten Emulator lauffähig sind (besser gesagt: sie laufen zwar auch unter dem normalen VT52-Emulator, weil dieser - Ausnahmen ausgenommen - die neuen ESC-Sequenzen einfach ignoriert), andererseits dürfte es wohl keine Schwierigkeiten bereiten, xVT52.PRG mit seinen 4.5 Kilobytes Speicherbedarf mit auf die entsprechende Programmdiskette zu kopieren, oder?

An den folgenden Escapes hat sich nichts geändert (d.h. sie unterscheiden sich nicht von denen des Originals):

- ESC A, B, C, D, E, H, I, J, K, L, M, und Y
- ESC d, e, f, l, o, p, q, v und w.

Im übrigen sind auch die normalen Steuerzeichen Carriage Return, Line Feed, VT, Backspace und Bell (Wau!) gleich geblieben.

Folgende Steuerzeichen weichen von der VT52-Norm ab:

- **ESC b und c:** Diese dienen dem Setzen der Vorder- und Hintergrundfarbe. Da xVT52 jedoch einen Monochrom-Monitor voraussetzt, werden beide Sequenzen schlichtweg ignoriert.
- **ESC j und k:** Mit der ersten Sequenz kann man die aktuelle Cursorposition speichern und mit ESC k wieder anspringen. Insofern sind sie zum Original kompatibel; allerdings gestatten sie nun das Speichern von bis zu drei (bisher eine) Positionen. Dabei wird mit ESC k immer die zuletzt gespeicherte Position zurückgeholt; ist keine gespeichert, so passiert auch nichts! Möchte man eine zurückgeholte Position später wieder anspringen, muß sie zuerst wieder gespeichert werden.

Von den gewöhnlichen Steuerzeichen hat sich TAB (0x9) geändert. Während die Tabulatorweite beim VT52 konstant 8 Zeichen beträgt, gestattet xVT52 das Setzen von bis zu 80 (!) Tabulatoren an beliebigen Spalten (zugegeben: bei Ausreizen der 80 Tabs ist die Beliebigkeit zumindest beim 80. natürlich etwas eingeschränkt!!). Standardmäßig ist jede achte Spalte mit einem Tabulator versehen, so daß man auch hier ohne weiteres zur Norm kompatibel bleiben kann, sofern man's möchte. Für den einfachen Aufbau von Tabellen ist es jedoch sicherlich nützlich, von den erweiterten Möglichkeiten Gebrauch zu machen. Hierzu stehen vier Escapes bereit:

- **ESC N** setzt auf jede achte Spalte einen Tabulator, was somit der VT52-Standardeinstellung entspricht.
- **ESC O** löscht alle Tabulatoren. Das kann sinnvoll sein, wenn man neue Tabulatoren setzen möchte und nicht weiß, wo bereits welche gesetzt sind, oder es mehr Aufwand bereiten würde, die bereits gesetzten und unerwünschten Tabs einzeln zu löschen.
- **ESC P n** setzt an Spalte n-32 (die erste Spalte ist 0) einen Tabulator. Möchte man also beispielsweise einen Tabulator auf Spalte 15 setzten, benutzt man z.B. in BASIC folgenden Befehl: Print chr$(27); "P";chr$(15+32);
- **ESC Q n** löscht einen an Spalte n-32 befindlichen Tabulator, wobei es keine Rolle spielt, ob dort vorher einer gesetzt war; will heißen es passiert nichts, wenn Sie einen nicht existierenden Tabulator löschen (warum sollte auch??!).

Wenn Sie sich über den Offset von 32 wundern sollten: auch hierfür hat's eine Erklärung. Wir reden hier die ganze Zeit von Emulatoren, als wären sie die natürlichsten Dinge der Welt neben Bananen und Bühnenlasern! Mal ganz ehrlich: Haben Sie 'ne Ahnung, warum es VT52-Emulator heißt? Geschweige denn, was man sich darunter vorzustellen hat? If not then read Historie else goto end.

Historie

Beginnen wir der Einfachheit halber wieder mit dem Ende: Ein Emulator ist die Simulation von Hardware mittels Software, d.h. ein Programm tut so, als ob, ist es aber gar nicht! So ist es ist beispielsweise möglich, den guten alten ST per Programm in einen noch guteren und alteren MacIntosh odergar in den allergutesten und alleraltesten Ih Bäh Mhh zu verwandeln, ohne auch nur ein Schräubchen berühren zu müssen. "Fein!", werden Sie jetzt sagen, "Und was hat das alles mit meinem VT52 zu tun?". Es hat. Sie müssen sich vorstellen, daß es Zeiten gab, in denen Computer sehr, sehr teuer waren. Anstatt also jedem Mitarbeiter einen Mega ST4 mit 60MB-Festplatte und Laserdrucker auf den Tisch zu stellen, gab es nur einen biederen grün leuchtenden Monitor, während sich der sogenannte Mainframe irgendwo zwischen Waschküche und Heizungskeller befand (zum einen, weil er so laut war, und zum anderen, weil die Warmluft der Röhren sich vortrefflich zum Wäschetrocknen eignete). Jaja, damals gab's halt noch keine Stiftung Staren-Nest und der Begriff "71Hertz" wurde noch vergeblich in Biologiebüchern gesucht. Zu dieser Zeit also trug es sich zu, daß man vor dem Problem stand, daß der Benutzer irgendwie mit dem Rechner in Verbindung treten mußte (darf ich's kommunizieren nennen?). Also baute man eine Hardware, nannte sie Terminal und befahl ihr, Dolmetscher zu spielen. In ihrem Innersten gab es eine Schaltung (heute würde man ja "Chip" sagen, aber damals gab es nur solche aus Kartoffeln), also etwas ureigen Elektrisches, was mittels eines Protokolls den Datenaustausch zwischen Tastatur, Mainframe und Monitor bewerkstelligte (Mäuse gab's ja nur im Heizungskeller...).

Natürlich ist die Problematik bis heute die gleiche geblieben; es gibt nach wie vor Zentralrechner, die eine bestimmte Anzahl von Terminals bedienen können. Nur werden die Protokolle nicht mehr nur über die Hardware realisiert - zunehmend übernehmen auch Programme diese

```
**************
* xVT52 DEMO *
**************

Zum Thema Schrift-Attribute: Es kommt relativ häufig vor, daß man
etwas unterstreichen möchte, um es optisch hervorzuheben.
Bisher mußte in solchen Fällen das umständlich und sehr aufwendig
zu programmierende VDI benutzt werden. Es stellte sich die Frage,
ob das so sein MUSS, und sie wurde mit ESC y beantwortet...

Genauso hat man jetzt die Möglichkeit, auf die halbhelle Schrift
umzuschalten. ESC R macht's möglich...

Es ist selbstredend, daß sich diese Attribute auch mischen lassen!
(Taste)
```

Verschiedene Schriftattribute

Aufgaben. Womit wir den Kreis geschlossen hätten, denn nun sind wir wieder beim Emulator. Bleibt immer noch die Frage nach dem ominösen Offset von 32. Er hängt halt zusammen mit dem erwähnten Protokoll (keine Angst, das ist nichts Unanständiges und gibt auch keine Flensburger!). Es regelt nämlich solch enorm wichtige Dinge wie etwa die Übertragungsgeschwindigkeit und Datensicherheit (Stichwort: Checksummenbildung), unterstützt aber auch besondere Funktionen, die mit speziellen Steuerzeichen ausgelöst werden (wie z.B. mit ESC, 0x1B). Damit nun aber unterschieden werden kann, was simpler Text ist und was nicht, hat man den Steuerzeichen einfach spezielle Codes zugeordnet. Beim ASCII-Code zum Bleistift liegen die Steuerzeichen im Bereich bis 0x1F, Zahlen, Buchstaben und Sonderzeichen zwischen 0x20 und 0x7F. Das genau ist der Grund, warum der Offset unbedingt notwendig ist - ohne ihn würde man sonst ungewollt ein Steuerzeichen abschicken, was schlimmstenfalls einen Abbruch der Datenübertragung zur Folge hätte! Das hätte ich Ihnen zwar alles viel kürzer verklikkern können, aber warum sollte ich mich bremsen, wenn's gerade gut aus der Feder läuft? Ich hoffe, Sie teilen meine Ansicht, nichts sei langweiliger als geballtes Wissen im DIN-Format; und vom Schmunzeln ist ja wohl auch noch keine Muskelfaser gerissen...

Zum Thema Textattribute stehen unter xVT52 folgende ESCapes zu Ihren Diensten:

- **ESC y:** Schaltet den Unterstreichungsmodus ein.
- **ESC z:** Schaltet den Unterstreichungsmodus aus.
- **ESC R:** Schaltet die halbhelle Darstellung ein.
- **ESC S:** Schaltet die halbhelle Darstellung aus.

Halbhelle Darstellung ist dabei gleichbedeutend mit dem Attribut "disabled" bei Drop-Down-Menüs und Dialogboxen. Die für diesen Effekt verantwortliche Funktion heißt LIGHTMASK und löscht für jede Scanline abwechselnd alle geraden und ungeraden Pixel, indem sie den ursprünglichen Inhalt mit 0x55 (%01010101) bzw. 0xAA (%10101010) verUNDet. Das Unterstreichen dagegen ist noch in der Routine PUT untergebracht, weil hier ja lediglich die unterste Scanline des Zeichens mit 0xFF zu ersetzen ist - sehr ökonomisch. Zwei weitere neue Sequenzen kümmern sich um den Font:

- **ESC F:** Schaltet auf den "normalen" 8x16-Font um.
- **ESC G:** Umschalten auf den "kleinen" 8x8-Font.

Hierbei ist zu erwähnen, daß beim Umschalten vom 8x16 in den 8x8-Font die Zeile verdoppelt und umgekehrt halbiert wird, damit die Cursorposition (zumindest bei geraden Zeilen) unverändert bleibt. Bereits zu Beginn des ersten Teils hatte ich ja bereits erwähnt, daß man mit ESC i die Ausgabeparameter zurücksetzen kann, so daß sich eine weitere Erklärung erübrigt. Ich verweise hier nochmal auf die Funktion INIT_CONOUT, die sich in des Listings erstem Teil befindet und aus der Sie ersehen können, was alles initialisiert wird. Da fast jede Zeile des Listings dokumentiert ist (was ich Ihnen für Assemblerprogramme nur wärmstens zur Nachahmung empfehlen kann!), hoffe ich, daß Sie mit dem Programm einigermaßen zurecht kommen - auch wenn Sie nicht gerade zu den Koryphäen des Motorola-Jargons zählen sollten...

```
TCB:
  dc.w 0       ; 0(TCB) aktuelle Spalte
  dc.w 0       ; 2(TCB) aktuelle Zeile
  dc.w 79      ; 4(TCB) maximale Spalte
  dc.w 24      ; 6(TCB) maximale Zeile
  dc.b 2       ; 8(TCB) Bitvektor für Attribute:
                     Bit 0=1: Invertieren ein (ESC p)
                     Bit 1=1: Wrapping ein (ESC v)
                     Bit 2=1: Unterstrich ein (ESC y)
                     Bit 3=1: Halbhell ein (ESC R)
  dc.b 0       ; 9(TCB) Grafikflag (ESC r; 0=aus, -1=ein)
  dc.w 0       ; 10(TCB) Anzahl gespeicherter Cursorpositionen (ESC j)
  dc.w 0,0     ; 12(TCB)  1. gespeicherte Position (x, y)
  dc.w 0,0     ; 16(TCB)  2. -"-
  dc.w 0,0     ; 20(TCB)  3. -"-
  dc.l 0       ; 24(TCB) Zeiger auf 8x16-Fontdaten (GEM)
  dc.l 0       ; 28(TCB) Zeiger auf 8x8-Fontdaten (GEM)
  dc.l 0       ; 32(TCB) Zeiger auf aktuellen Font
  dc.w 16      ; 36(TCB) Höhe eines Zeichens in Pixel
  dc.w 16*80   ; 38(TCB) Bytes pro Textzeile
```

Tabelle 1a: Aufbau der Variablen TCB (TerminalControlBlock)

Damit's für diesmal nicht zu lang wird, möchte ich an dieser Stelle einen Breakpoint setzen; jedoch nicht ohne Ihnen beim Abtippen viel Spaß und wenig Krämpfe gewünscht und Sie darauf hingewiesen zu haben, was Sie im dritten Teil erwartet: Einerseits die weitere Besprechung der neuen ESC-Sequenzen, andererseits will ich Ihnen verraten, wie man zeitkritische Programme auf Trab bringen kann und - naja, mal sehen... Alsdann tschüß bis neulich!

MS

```
CCB:
  dc.b 2      ; 0(CCB) Bitvektor für Cursor-Status:
                  Bit 0=1: Cursor enable (für Interrupt!)
                  Bit 1=1: Cursor darf blinken
                  Bit 2=1: Cursorposition invertiert
                  Bit 3=1: Cursor eingeschaltet
  dc.b 0      ; 1(CCB) Anzahl gespeicherter CUR_OFFs (ESC f)
  dc.w 20     ; 2(CCB) Blinkrate
  dc.w 20     ; 4(CCB) Zähler für Blinken (für Interrupt)
  dc.l 0      ; 6(CCB) absolute Cursorposition (im Video-RAM)
```

Tabelle 1b: Aufbau der Variablen CCB (CursorControlBlock)

```
  1: PUT:                            ; Zeichen in d0.w ausgeben
  2:                                 ; (verändert: a0-a4/d0-d3)
  3:   lea     CCB,a4                ; ^Cursor-Controlblock
  4:   bclr    #0,(a4)               ; disable Cursor
  5:   bclr    #2,(a4)               ; Cursorposition nicht
                                       invertiert
  6:   lea     TCB,a0                ; ^TerminalControlBlock
  7:   move.l  32(a0),a1             ; ^aktuelle Fontdaten
  8:   lea     \wr_16,a3             ; (16 Scan-Zeilen ausgeben)
  9:   cmp.l   24(a0),a1             ; aktueller Font=8x16?
 10:   beq.s   \write                ; ja
 11:   lea     \wr_end,a3            ; sonst nur 8 Scan-Zeilen
                                       ausgeben
 12: \write:
 13:   adda.w  d0,a1                 ; Offset addieren
 14:   move.l  6(a4),a2              ; abs. Cursorposition
 15:   move.b  (a1),(a2)+            ; Scan-Zeilen übertragen;
                                       Spalte ++
 16:   move.b  01*256(a1),01*80-1(a2)
 17:   move.b  02*256(a1),02*80-1(a2)
 18:   move.b  03*256(a1),03*80-1(a2)
 19:   move.b  04*256(a1),04*80-1(a2)
 20:   move.b  05*256(a1),05*80-1(a2)
 21:   move.b  06*256(a1),06*80-1(a2)
 22:   move.b  07*256(a1),07*80-1(a2) ; letzte Scan-Zeile 8x8
 23:   jmp     (a3)                   ; Je nachdem, ob 8x8
                                        oder 8x16-Font gewählt
 24: \wr_16:                          ; ist, \wr_end oder
                                        \wr_16 anspringen
 25:   move.b  08*256(a1),08*80-1(a2)
 26:   move.b  09*256(a1),09*80-1(a2)
 27:   move.b  10*256(a1),10*80-1(a2)
 28:   move.b  11*256(a1),11*80-1(a2)
 29:   move.b  12*256(a1),12*80-1(a2)
 30:   move.b  13*256(a1),13*80-1(a2)
 31:   move.b  14*256(a1),14*80-1(a2)
 32:   move.b  15*256(a1),15*80-1(a2) ; letzte Scan-Zeile 8x16
 33: \wr_end:
 34:   btst    #2,8(a0)              ; Unterstrich ein?
 35:   beq.s   \half                 ; nein
 36:   moveq   #80,d1                ; Bytes/Pixelzeile
 37:   move.w  36(a0),d2             ; Zeichenhöhe in Pixel
 38:   subq.w  #1,d2                 ; -1
 39:   mulu    d2,d1                 ; Offset bestimmen
 40:   move.b  #$FF,-1(a2,d1.w)      ; letzte Scan-Zeile
                                       unterstreichen
 41: \half:
 42:   btst    #3,8(a0)              ; halbe Helligkeit ein?
 43:   beq.s   \inv                  ; nein
 44:   bsr     LIGHTMASK             ; sonst maskieren
 45: \inv:
 46:   btst    #0,8(a0)              ; inverse Darstellung?
 47:   beq.s   \inc_col              ; nein
 48:   bsr     CUR_INV               ; sonst invertieren
 49:   bchg    #2,(a4)               ; alten Cursor-Status
                                       restaurieren
 50: \inc_col:
 51:   moveq   #80,d1                ; Bytes/Pixelzeile
 52:   move.l  a2,6(a4)              ; abs. Cursorposition++
 53:   addq.w  #1,(a0)               ; Spalte inkrementieren
 54:   cmp.w   (a0),d1               ; letzte Spalte erreicht?
 55:   bgt.s   \return               ; nein, fertig
 56:   btst    #1,8(a0)              ; wrap eingeschaltet?
 57:   bne.s   \wrap                 ; ja
 58:   subq.w  #1,(a0)               ; sonst letzte Spalte
 59:   subq.l  #1,6(a4)              ; und abs. Cursorposition
                                       beibehalten
 60:   bra.s   \return               ; Ende
 61: \wrap:
 62:   clr.w   (a0)                  ; Spalte 0
 63:   move.w  38(a0),d3             ; Bytes/Textzeile
 64:   ext.l   d3                    ; auf Langwort bringen
 65:   add.l   d3,6(a4)              ; + abs. Cursorposition
 66:   sub.l   d1,6(a4)              ; - 1 Pixelzeile = neue
                                       Position
 67:   addq.w  #1,2(a0)              ; Zeile ++
 68:   move.w  6(a0),d2              ; letzte Zeile
 69:   cmp.w   2(a0),d2              ; überschritten?
 70:   bpl.s   \return               ; nein, fertig
 71:   subq.w  #1,2(a0)              ; sonst letzte Zeile
                                       beibehalten
 72:   sub.l   d3,6(a4)              ; ebenso abs. Position
 73:   bsr     SCROLL_UP             ; alles 1 Zeile hochschieben
 74:   bsr     DEL_LINE              ; und letzte Zeile löschen
 75: \return:
 76:   bset    #0,(a4)               ; enable Cursor
 77:   clr.l   d0                    ; "kein Fehler"
 78:   rts                           ; tschau
 79:
 80: CUR_INV:                        ; Cursorposition invertieren
 81:                                 ; (verändert: d1/a1-a2/a4)
 82:   move.l  a2,-(a7)              ; Register retten
 83:   lea     TCB,a0                ; ^TerminalControlBlock
 84:   lea     CCB,a4                ; ^CursorControlBlock
 85:   move.l  6(a4),a1              ; abs. Cursor-Adresse
 86:   lea     \inv_end,a2           ; 8*8 annehmen
 87:   move.w  36(a0),d1             ; Zeichenhöhe
 88:   subq.w  #8,d1                 ; -Offset für 8*8
 89:   beq.s   \inv                  ; ok, 8*8-Font
 90:   lea     \inv_16,a2            ; sonst 16 Bytes NOTten
 91: \inv:
 92:   not.b   (a1)
 93:   not.b   0080(a1)
 94:   not.b   0160(a1)
 95:   not.b   0240(a1)
 96:   not.b   0320(a1)
 97:   not.b   0400(a1)
 98:   not.b   0480(a1)
 99:   not.b   0560(a1)
100:   jmp     (a2)
101: \inv_16:
102:   not.b   0640(a1)
103:   not.b   0720(a1)
104:   not.b   0800(a1)
105:   not.b   0880(a1)
106:   not.b   0960(a1)
107:   not.b   1040(a1)
108:   not.b   1120(a1)
109:   not.b   1200(a1)
110: \inv_end:
111:   bchg    #2,(a4)               ; Cursorposition invertiert
112:   move.l  (a7)+,a2              ; Register zurück
113:   rts
114:
115: LIGHTMASK:                      ; Zeichen unter Cursor
                                       halbhell darstellen
116:   movem.l d3/d4/a2,-(a7)        ; Register retten
117:   lea     TCB,a0                ; ^TerminalControlBlock
118:   lea     CCB,a4                ; ^CursorControlBlock
119:   move.w  #$AA,d3               ; Bitmaske für gerade
                                       Pixelzeilen
120:   moveq   #$55,d4               ; Bitmaske für ungerade
                                       Pixelzeilen
121:   move.l  6(a4),a1              ; abs. Cursor-Adresse
122:   lea     \light_end,a2         ; 8*8 annehmen
123:   move.w  36(a0),d1             ; Zeichenhöhe
124:   subq.w  #8,d1                 ; -Offset für 8*8
125:   beq.s   \light                ; ok, 8*8-Font
126:   lea     \light_16,a2          ; sonst 16 Bytes maskieren
127: \light:
128:   and.b   d3,(a1)
129:   and.b   d4,0080(a1)
```

```
130:  and.b    d3,0160(a1)
131:  and.b    d4,0240(a1)
132:  and.b    d3,0320(a1)
133:  and.b    d4,0400(a1)
134:  and.b    d3,0480(a1)
135:  and.b    d4,0560(a1)
136:  jmp      (a2)
137: \light_16:
138:  and.b    d3,0640(a1)
139:  and.b    d4,0720(a1)
140:  and.b    d3,0800(a1)
141:  and.b    d4,0880(a1)
142:  and.b    d3,0960(a1)
143:  and.b    d4,1040(a1)
144:  and.b    d3,1120(a1)
145:  and.b    d4,1200(a1)
146: \light_end:
147:  movem.l  (a7)+,d3/d4/a2   ; Register zurück
148:  rts
149:
150: DEL_SCRUP:                 ; ESC 'M'
151:  move.w   6(a0),d1         ; max. Zeile
152:  sub.w    2(a0),d1         ; -akt. Zeile=Anzahl zu
                                  scrollender Zeilen
153:  beq.s    \del             ; schon in letzter Zeile, nur
                                  noch löschen
154:  move.l   6(a4),a1         ; abs. Cursorposition
155:  suba.w   (a0),a1          ; -akt. Spalte=^Zeilenanfang
156:  bsr.s    SCR_UP_ENTRY     ; hochscrollen
157: \del:
158:  move.w   6(a0),d3         ; letzte Zeile
159:  bra      DEL_ENTRY        ; löschen und zurückkehren
160:
161: SCROLL_UP:                 ; Bildschirminhalt incl. der
                                  akt. Zeile
162:                            ; um eine Zeile nach oben
                                  schieben
163:  move.l   LOGBASE,a1       ; ^Video-RAM
164:  move.w   2(a0),d1         ; aktuelle Zeile
165:  beq      SCR_UP_END       ; nicht scrollen, falls schon
                                  oben
166: SCR_UP_ENTRY:              ; Einsprung für ESC 'M'
167:  subq.w   #1,d1            ; d1 in dbra-Zähler wandeln
168:  move.w   38(a0),d0        ; Bytes/Textzeile
169:  move.w   36(a0),d2        ; Zeichenhöhe
170:  lsr.w    #2,d2            ; div 4 (es werden immer 4
                                  Pixelzeilen verschoben)
171:  subq.w   #1,d2            ; in dbra-Zähler wandeln
172:  move.l   a1,a2            ; Zieladresse (a1)
173:  adda.w   d0,a2            ; + Bytes/Textzeile =
                                  Quelladresse (a2)
174:  movem.l  a0/a4/a6,-(a7)   ; Register retten
175: \scr_lns:                  ; d1+1 Zeilen nach oben
                                  scrollen
176:  move.w   d2,d0            ; Zeichenhöhe/4-1
177: \scr_ln:                   ; 1 Textzeile nach oben
                                  scrollen
178:  movem.l  (a2)+,REGISTER
179:  movem.l  REGISTER,(a1)
180:  movem.l  (a2)+,REGISTER
181:  movem.l  REGISTER,40(a1)  ; 4 komplette Pixelzeilen
                                  verschieben
182:  movem.l  (a2)+,REGISTER
183:  movem.l  REGISTER,80(a1)
184:  movem.l  (a2)+,REGISTER
185:  movem.l  REGISTER,120(a1)
186:  movem.l  (a2)+,REGISTER
187:  movem.l  REGISTER,160(a1)
188:  movem.l  (a2)+,REGISTER
189:  movem.l  REGISTER,200(a1)
190:  movem.l  (a2)+,REGISTER
191:  movem.l  REGISTER,240(a1)
192:  movem.l  (a2)+,REGISTER
193:  movem.l  REGISTER,280(a1)
194:  adda.w   #320,a1
195:  dbra     d0,\scr_ln       ; die nächsten 4 Pixelzeilen
                                  verschieben
196:  dbra     d1,\scr_lns      ; nächste Textzeile
197:  movem.l  (a7)+,a0/a4/a6   ; Register zurück
198: SCR_UP_END:
199:  rts
200:
201: INS_SCRDN:                 ; ESC 'L'
202:  bsr.s    SCROLL_DOWN      ; Zeile einfügen
203:  bra      DEL_LINE         ; und löschen
204:
205: DEL_SCRDN:                 ; ESC 't'
206:  move.w   2(a0),d1         ; akt. Zeile
207:  beq.s    \del             ; schon oben, nur noch löschen
208:  move.l   6(a4),a1         ; abs. Cursorposition
209:  suba.w   (a0),a1          ; -akt. Spalte=^Zeilenanfang
210:  adda.w   38(a0),a1        ; +Bytes/Textzeile
                                  (1 Textzeile tiefer)
211:  suba.w   #80,a1           ; -80 Bytes (1 Pixelzeile
                                  höher)
212:  bsr.s    SCR_DN_ENTRY     ; runterscrollen
213: \del:
214:  clr.w    d3               ; oberste Zeile (0)
215:  bra      DEL_ENTRY        ; löschen und zurückkehren
216:
217: SCROLL_DOWN:               ; Bildschirminhalt incl. der
                                  akt. Zeile
218:                            ; um eine Zeile nach unten
                                  schieben
219:  move.l   LOGBASE,a1       ; ^Video-RAM
220:  adda.l   #32000-80,a1     ; ^letzte Pixel-Zeile
221:  move.w   6(a0),d1         ; max. Zeile
222:  sub.w    2(a0),d1         ; -aktuelle=Anzahl zu
                                  scrollender Zeilen
223:  beq      SCR_DN_END       ; nicht scrollen, falls schon
                                  unten
224: SCR_DN_ENTRY:              ; Einsprung für ESC 't'
225:  subq.w   #1,d1            ; sonst in dbra-Zähler wandeln
226:  move.w   38(a0),d0        ; Anzahl Bytes/Textzeile =
                                  Offset
227:  move.w   36(a0),d2        ; Zeichenhöhe
228:  lsr.w    #2,d2            ; div 4 (es werden immer 4
                                  Pixelzeilen verschoben)
229:  subq.w   #1,d2            ; in dbra-Zähler wandeln
230:  move.l   a1,a2            ; Zieladresse (a1)
231:  suba.w   d0,a2            ; - Offset = Quelladresse (a2)
232:  movem.l  a0/a4/a6,-(a7)   ; Register retten
233: \scr_lns:                  ; d1+1 Zeilen nach unten
                                  scrollen
234:  move.w   d2,d0            ; Zeichenhöhe/4-1
235: \scr_ln:                   ; 1 Textzeile nach unten
                                  scrollen
236:  movem.l  (a2)+,REGISTER
237:  movem.l  REGISTER,(a1)
238:  movem.l  (a2),REGISTER
239:  movem.l  REGISTER,40(a1)  ; 4 komplette Pixelzeilen
                                  verschieben
240:  suba.w   #120,a2
241:  movem.l  (a2)+,REGISTER
242:  movem.l  REGISTER,-80(a1)
243:  movem.l  (a2),REGISTER
244:  movem.l  REGISTER,-40(a1)
245:  suba.w   #120,a2
246:  movem.l  (a2)+,REGISTER
247:  movem.l  REGISTER,-160(a1)
248:  movem.l  (a2),REGISTER
249:  movem.l  REGISTER,-120(a1)
250:  suba.w   #120,a2
251:  movem.l  (a2)+,REGISTER
252:  movem.l  REGISTER,-240(a1)
253:  movem.l  (a2),REGISTER
254:  movem.l  REGISTER,-200(a1)
255:  suba.w   #120,a2
256:  suba.w   #320,a1
257:  dbra     d0,\scr_ln       ; die nächsten 4 Pixelzeilen
                                  verschieben
258:  dbra     d1,\scr_lns      ; nächste Textzeile
259:  movem.l  (a7)+,a0/a4/a6   ; Register zurück
260: SCR_DN_END:
261:  rts
262:
263: DEL_LINE:                  ; akt. Zeile löschen, Cursor
                                  auf Spalte 0
264:  move.w   2(a0),d3         ; akt. Zeile
265: DEL_ENTRY:                 ; Einsprung für Löschen
                                  beliebiger Zeilen
266:  move.l   LOGBASE,a1       ; ^Video-RAM
267:  addq.w   #1,d3            ; ++
268:  move.w   38(a0),d1        ; Bytes/Textzeile
269:  mulu     d3,d1            ; (akt. Zeile+1)*
                                  (Bytes/Textzeile)
270:  adda.l   d1,a1            ; = Bildschirm-Offset
271:  move.w   36(a0),d2        ; Zeichenhöhe
272:  lsr.w    #2,d2            ; div 4 (es werden immer 4
                                  Pixelzeilen gelöscht)
273:  subq.w   #1,d2            ; in dbra-Zähler wandeln
274:  movem.l  a0/a4/a6,-(a7)   ; Register retten
275:  movem.w  ZEROES,REGISTER  ; Register löschen
276: \lp:
277:  movem.l  REGISTER,-(a1)   ; 4 komplette Pixelzeilen
                                  löschen
278:  movem.l  REGISTER,-(a1)
279:  movem.l  REGISTER,-(a1)
280:  movem.l  REGISTER,-(a1)
281:  movem.l  REGISTER,-(a1)
282:  movem.l  REGISTER,-(a1)
283:  movem.l  REGISTER,-(a1)
284:  movem.l  REGISTER,-(a1)
285:  dbra     d2,\lp           ; nächste Pixelzeile
286:  movem.l  (a7)+,a0/a4/a6   ; Register zurück
287:  clr.w    (a0)             ; Cursor in Spalte 0
288:  rts                       ; fertig
289:
290:
291: BEL:                       ; Control G (bingggg!)
```

```
292:   suba.l   a5,a5             ; A5 löschen
293:   move.l   BEL_ADR,-(a7)     ; Original TOS-Routine
294:   rts                        ; anspringen
295:
296:  CRS_LEFT:                   ; ESC 'C' (=Backspace BS)
297:  BS:
298:   tst.w    (a0)              ; Spalte 0?
299:   beq.s    \zurück           ; ja, BS nicht möglich
300:   subq.w   #1,(a0)           ; sonst Spalte--
301:  \zurück:
302:   rts                        ; das war's
303:
304:  CALC_POS:                   ; nächste Tab-Position
                                    berechnen
305:   clr.l    d7                ; Default-Rückgabewert: Kein
                                    Fehler
306:   move.w   d0,d1             ; Spalte retten für Byte-Offset
307:   move.w   d0,d2             ; nochmal retten für Bit-Offset
308:   moveq    #7,d3             ; Bit-Zähler initialisieren
309:   lsr.w    #3,d1             ; Byteoffset berechnen
                                    (Spalte div 8)
310:   and.w    d3,d2             ; Divisionsrest (Spalte mod 8)
311:   sub.w    d2,d3             ; von 7 subtrahiert =
                                    Bit-Offset
312:  \bit_lp:
313:   btst     d3,0(a1,d1.w)     ; Tab gesetzt?
314:   beq.s    \weiter           ; nein
315:   rts                        ; sonst zurück
316:  \weiter:
317:   addq.w   #1,d0             ; Spalte ++
318:   dbra     d3,\bit_lp        ; und weitersuchen
319:  \byte_lp:
320:   addq.w   #1,d1             ; nächstes Byte des
                                    Tabulatoren-Bit-Vektors
321:   cmpi.w   #10,d1            ; schon letztes Byte
                                    überschritten?
322:   bge.s    \fail             ; ja, mit Fehler zurück
323:   tst.b    0(a1,d1.w)        ; gibt's hier'n Tab?
324:   bne.s    \s_next           ; yeah!
325:   addq.w   #8,d0             ; sonst Spalte +=
                                    8 (!@$%"?C-Syntax!)
326:   bra.s    \byte_lp          ; und nächstes Byte untersuchen
327:  \s_next:
328:   moveq    #7,d3             ; wieder beim MSB mit der
329:   bra.s    \bit_lp           ; Suche beginnen
330:  \fail:
331:   moveq    #-1,d7            ; nicht fündig geworden...
332:   rts                        ; und tschüß
333:
334:  TAB:
335:   lea      TABS,a1           ; ^TabulatorenBitVektor
336:   move.w   (a0),d0           ; akt. Spalte
337:   addq.w   #1,d0             ; ++
338:   cmpi.w   #80,d0            ; >=letzte?
339:   bge.s    \wrap             ; ja, auf Wrap testen
340:   bsr.s    CALC_POS          ; sonst nächsten Tab suchen
341:   tst.l    d7                ; gefunden?
342:   bmi.s    \wrap             ; nö
343:   move.w   d0,(a0)           ; sonst Spalte setzen
344:  \ende:
345:   rts                        ; und zurück
346:  \wrap:
347:   btst     #1,8(a0)          ; Wrapping erlaubt?
348:   beq.s    \ende             ; nein, nichts ändern
349:   clr.w    d0                ; sonst ersten
350:   bsr.s    CALC_POS          ; Tabulator finden
351:   tst.l    d7                ; keiner da?
352:   bmi.s    \ende             ; letztendlich nicht, fertig
353:   move.w   d0,(a0)           ; sonst Spalte setzen und LF
                                    ausführen
354:
355:  LF:
356:   move.w   2(a0),d0          ; akt. Zeile
357:   cmp.w    6(a0),d0          ; =letzte?
358:   bmi.s    \ok               ; nein
359:   bsr      SCROLL_UP         ; sonst scrollen
360:   move.w   (a0),-(a7)        ; akt. Spalte merken
361:   bsr      DEL_LINE          ; und letzte Zeile löschen
362:   move.w   (a7)+,(a0)        ; akt. Spalte zurück
363:   rts
364:  \ok:
365:   addq.w   #1,2(a0)          ; Zeile++
366:   rts
367:
368:  CR:
369:   clr.w    (a0)              ; Cursor in Spalte 0 setzen
370:   rts                        ; fertig
371:
372:
373:  CLS:                        ; Bildschirm löschen+Home
374:   movem.l  a0/a6,-(a7)       ; Register retten
375:   moveq    #49,d0            ; 50 mal 8 Pixelzeilen löschen
376:   move.l   LOGBASE,a6        ; ^Video-RAM
377:   adda.w   #32000,a6         ; +32000=Bildschirmende
378:  CLS_ENTRY:                  ; Entry point für teilweises
                                    Löschen
379:   movem.w  ZEROES,d1-a2      ; 10 Register löschen
380:  \lp:
381:   movem.l  d1-a2,-(a6)       ; 8 komplette Pixelzeilen
                                    löschen
382:   movem.l  d1-a2,-(a6)       ; (jeder movem-Befehl löscht
                                    40 Bytes)
383:   movem.l  d1-a2,-(a6)
384:   movem.l  d1-a2,-(a6)
385:   movem.l  d1-a2,-(a6)
386:   movem.l  d1-a2,-(a6)
387:   movem.l  d1-a2,-(a6)
388:   movem.l  d1-a2,-(a6)
389:   movem.l  d1-a2,-(a6)
390:   movem.l  d1-a2,-(a6)
391:   movem.l  d1-a2,-(a6)
392:   movem.l  d1-a2,-(a6)
393:   movem.l  d1-a2,-(a6)
394:   movem.l  d1-a2,-(a6)
395:   movem.l  d1-a2,-(a6)
396:   movem.l  d1-a2,-(a6)
397:   dbra     d0,\lp            ; die nächste 10 Zeilen
398:   movem.l  (a7)+,a0/a6       ; Register zurück
399:
400:  HOME:                       ; Cursor nach 0,0
401:   clr.w    (a0)              ; Spalte 0
402:   clr.w    2(a0)             ; Zeile 0
403:   rts
404:
405:
406:  E_BIG:                      ; Offset-Tabelle für ESC
                                    "Großbuchstabe"
407:   dc.w 0                     ; A Cursor rauf
408:   dc.w CRS_DOWN-CRS_UP       ; B Cursor runter
409:   dc.w CRS_RIGHT-CRS_UP      ; C Cursor rechts
410:   dc.w CRS_LEFT-CRS_UP       ; D Cursor links
411:   dc.w CLS-CRS_UP            ; E Bildschirm löschen
412:   dc.w BIG_FONT-CRS_UP       ; F 8x16-Font aktivieren
413:   dc.w SML_FONT-CRS_UP       ; G 8x8-Font aktivieren
414:   dc.w HOME-CRS_UP           ; H Home
415:   dc.w CRS_UP_SCR-CRS_UP     ; I Cursor rauf (Scrollen,
                                    falls in Zeile 0)
416:   dc.w DEL_FROM_CRS-CRS_UP   ; J Löschen ab Cursor bis
                                    Bildschirm-Ende
417:   dc.w L_TO_END-CRS_UP       ; K Löschen ab Cursor bis
                                    Zeilenende
418:   dc.w INS_SCRDN-CRS_UP      ; L Zeile einfügen, Rest
                                    runterschieben
419:   dc.w DEL_SCRUP-CRS_UP      ; M Zeile löschen, Rest
                                    hochscrollen
420:   dc.w DEF_TABS-CRS_UP       ; N Tabulatoren an jede
                                    8. Spalte setzen
421:   dc.w CLR_TABS-CRS_UP       ; O alle Tabulatoren löschen
                                    (Tabula rasa)
422:   dc.w SET_TAB-CRS_UP        ; P Tabulator setzen
423:   dc.w CLR_TAB-CRS_UP        ; Q Tabulator löschen
424:   dc.w LIGHT_ON-CRS_UP       ; R halbe Helligkeit
                                    einschalten
425:   dc.w LIGHT_OFF-CRS_UP      ; S volle Helligkeit
                                    einschalten
426:   dc.w FSCR_LEFT-CRS_UP      ; T akt. Zeile um 1 Pixel
                                    nach links scrollen
427:   dc.w FSCR_RIGHT-CRS_UP     ; U akt. Zeile um 1 Pixel
                                    nach rechts scrollen
428:   dc.w FSCRDN-CRS_UP         ; V incl. akt. Zeile
                                    pixelweise n. unten scrollen
429:   dc.w FDSCRUP-CRS_UP        ; W akt. Zeile löschen, Rest
                                    pixelweise hochscrollen
430:   dc.w FSCRUP-CRS_UP         ; X incl. akt. Zeile
                                    pixelweise n. oben scrollen
431:   dc.w DIR_CRS-CRS_UP        ; Y Cursor positionieren
                                    (Zeile, Spalte)
432:   dc.w FDSCRDN-CRS_UP        ; Z akt. Zeile löschen, Rest
                                    pixelweise runter
433:
434:
435:  E_SML:                      ; Offset-Tabelle für ESC
                                    "Kleinbuchstabe"
436:   dc.w CRS_ON_T-SET_COLOR    ; a Cursor einschalten, falls
                                    ESC 'f'-Zähler=0
437:   dc.w 0                     ; b Vordergrundfarbe setzen
438:   dc.w 0                     ; c Hintergrundfarbe setzen
439:   dc.w DEL_TO_CRS-SET_COLOR; d von Bildschirmanfang bis
                                    incl. Cursor löschen
440:   dc.w CRS_ON-SET_COLOR      ; e Cursor einschalten, ESC
                                    'f'-Zähler löschen
441:   dc.w CRS_OFF-SET_COLOR     ; f Cursor ausschalten
442:   dc.w GET_TCB-SET_COLOR     ; g Pointer auf
                                    TerminalControlBlock holen
443:   dc.w GET_CCB-SET_COLOR     ; h Pointer auf
                                    CursorControlBlock holen
444:   dc.w RESETCO-SET_COLOR     ; i Terminal initialisieren
445:   dc.w SAVE_CRS-SET_COLOR    ; j Cursorposition speichern
446:   dc.w REST_CRS-SET_COLOR    ; k Cursor an zuletzt
                                    gemerkte Stelle
```

```
447:    dc.w DEL_LINE-SET_COLOR  ; l Zeile löschen
448:    dc.w WHITE_BLK-SET_COLOR ; m weiße Schrift auf
                                   schwarzem Grund
449:    dc.w BLK_WHITE-SET_COLOR ; n schwarze Schrift auf
                                   weißem Grund
450:    dc.w L_TO_CRS-SET_COLOR  ; o von Zeilenanfang bis incl.
                                   Cursor löschen
451:    dc.w INV_ON-SET_COLOR    ; p inverse Darstellung ein
452:    dc.w INV_OFF-SET_COLOR   ; q inverse Darstellung aus
453:    dc.w GRAPHMODE-SET_COLOR ; r Grafikmodus
454:    dc.w INS_SCRUP-SET_COLOR ; s Zeile einfügen, Rest
                                   hochschieben
455:    dc.w DEL_SCRDN-SET_COLOR ; t Zeile löschen, Rest
                                   runterschieben
456:    dc.w SCR_LEFT-SET_COLOR  ; u akt. Zeile um 1 Spalte
                                   nach links scrollen
457:    dc.w WRAP_ON-SET_COLOR   ; v automatischen
                                   Zeilenüberlauf einschalten
458:    dc.w WRAP_OFF-SET_COLOR  ; w automatischen
                                   Zeilenüberlauf ausschalten
459:    dc.w SCR_RIGHT-SET_COLOR ; x akt. Zeile um 1 Spalte
                                   nach rechts scrollen
460:    dc.w UNDER_ON-SET_COLOR  ; y Unterstrich ein
461:    dc.w UNDER_OFF-SET_COLOR ; z Unterstrich aus
462:
463:
464:
465:  CRS_ON_T:                  ; ESC 'a'
466:    tst.b    1(a4)           ; liegt kein ESC 'f'-Aufruf
                                   vor?
467:    beq.s    \ein            ; nein, einschalten
468:    subq.b   #1,1(a4)        ; sonst Anzahl der Aufrufe
                                   dekrementieren
469:    bne.s    \zurück         ; nur bei 0 einschalten
470:  \ein:
471:    bset     #3,(a4)         ; Cursor einschalten
472:  \zurück:
473:    rts
474:
475:  SET_COLOR:                 ; ESC 'b'/'c' (nicht
                                   unterstützt)
476:    lea      VEC_BASE,a1     ; Vektor
477:    lea      SET_COLOR2,a2   ; auf Holen des Farbwertes
478:    move.l   a2,(a1)         ; umschalten
479:    rts
480:  SET_COLOR2:
481:    lea      VEC_BASE,a1     ; Vektor wieder
482:    lea      STD_VEC,a2      ; auf normale Ausgabe
483:    move.l   a2,(a1)         ; umschalten
484:    rts
485:
486:  CRS_ON:                    ; ESC 'e'
487:    bset     #3,(a4)         ; Cursor einschalten
488:    clr.b    1(a4)           ; Anzahl CRS_OFF löschen
489:    rts
490:
491:  CRS_OFF:                   ; ESC 'f'
492:    bclr     #3,(a4)         ; Cursor ausschalten
493:    addq.b   #1,1(a4)        ; Anzahl der Aufrufe
                                   inkrementieren
494:    rts
495:
496:  INV_ON:                    ; ESC 'p'
497:    bset     #0,8(a0)        ; inverse Darstellung ein
498:    rts
499:
500:  INV_OFF:                   ; ESC 'q'
501:    bclr     #0,8(a0)        ; inverse Darstellung aus
502:    rts
503:
504:  WRAP_ON:                   ; ESC 'v'
505:    bset     #1,8(a0)        ; Wrapping einschalten
506:    rts
507:
508:  WRAP_OFF:                  ; ESC 'w'
509:    bclr     #1,8(a0)        ; Wrapping ausschalten
510:    rts
511:
512:  UNDER_ON:                  ; ESC 'y'
513:    bset     #2,8(a0)        ; Unterstrich einschalten
514:    rts
515:
516:  UNDER_OFF:                 ; ESC 'z'
517:    bclr     #2,8(a0)        ; Unterstrich ausschalten
518:    rts
519:
520:  INS_SCRUP:                 ; ESC 's'
521:    bsr      SCROLL_UP       ; Zeile einfügen, Rest
                                   hochschieben
522:    bsr      DEL_LINE        ; und neue Zeile löschen
523:    rts
```

ENDE



In der diesmaligen Folge unserer Programmierpraxis geht es wieder voll durch alle Sprachen durch. Alle ist natürlich etwas übertrieben, aber für die Freunde von C, Modula-2 und BASIC ist diesmal etwas dabei.

Mit der Harddisk-Info kann man sich die Aufteilung einer Festplatte anzeigen lassen. Ausgegeben werden Gesamtgröße der Festplatte, wieviel KBytes belegt und wieviel frei sind, wieviel defekte Sektoren markiert wurden und ...

Wer einmal auf dem MacIntosh gearbeitet hat, kennt die praktische Möglichkeit, nur den Inhalt eines Fenster auf dem Drucker auszugeben. Damit auch die ATARI ST-Benutzer ihr Papier nicht mehr nur überwiegend mit dem Graumuster des Desktops füllen müssen, wurde das Accessory PRINTTOP geschrieben.

In compiliertem GFA-BASIC wird beim Laden des Programm-Codes grundsätzlich der Bildschirm gelöscht. Nun ist dies aber besonders bei Programmen, die mit einer GEM-Umgebung arbeiten, nicht gerade wünschenswert. Ein kleines Programm ändert deshalb ein compiliertes Programm so ab, daß der Bildschirm nach dem Laden erhalten bleibt.

Das hier vorgestellte Listing zeigt Ihnen, wie Sie eine Uhr in der Sprache C selbst programmieren und die selbst überwachte Uhrzeit auf dem Bildschirm anzeigen können, ohne auf vorgefertigte Uhr- und Ausgabefunktionen zurückzugreifen. Neben anderen nützlichen Informationen erfahren Sie außerdem, wie eigene Funktionen in die VBL-Slots eingebunden und wie Systemvariablen benutzt werden.

Zu guter Letzt haben wir noch eine Shell, also ein Manager-programm, zum Verwalten von BASIC-Interpreter und Compiler. Man erspart sich das lästige Doppelklicken im Desktop, wenn man bei der Programmentwicklung mit den beiden arbeitet. Angewandt wird dabei die Pexec-Routine.

I N H A L T

IMMER AUF DEM LAUFENDEN

Was belegt wie Ihre Festplatte

Festplatten als Massenspeicher erfreuen sich, besonders bei ST-Anwendern, wachsender Beliebtheit. Nachdem ein 20-Megabyte-Schuhkarton mit der Aufschrift SH204 sich viel Kritik über Lautstärke, Aussehen und kurze Kabel gefallen lassen mußte, hat Atari die SH205 gebaut, die sehr gut in das Design der Mega-ST-Reihe paßt. Bei solchen Speicherkapazitäten ist es wichtig, stets den noch freien Platz auf der Festplatte zu wissen. Das folgende Programm gibt Antworten auf folgende Fragen aus:

- **Wie groß ist die gesamte Platte ?**
- **Wieviel davon ist belegt, wieviel ist noch frei ?**
- **Wieviele defekte Sektoren wurden markiert ?**
- **Wie groß sind die einzelnen Partitions und wieviel Platz ist belegt?**

Dazu benutzt die 'HARDDISK-INFORMATION' die Information im Bootsektor der Festplatte und die GEMDOS-Funktion Dfree, die den freien Platz auf einem Massenspeichermedium berechnet. Analog zum Bootsektor einer Diskette enthält der Bootsektor die vom Programm benötigte Information über Größe der Platte sowie Vorhandensein und Größen der einzelnen Partitionen. Der Bootsektor ist - wie bei den Disketten - der erste physikalische Sektor auf der Festplatte. Sein Aufbau ist im Listing in der Struktur BOOTBLOCK definiert.

Das Programm muß also den Bootsektor lesen, um dessen Inhalt auswerten zu können. Ein Problem besteht in der Tatsache, daß der Bootsektor einer Festplatte keinesfalls identisch ist mit dem ersten Sektor der ersten Partition (vom Benutzer meist als Laufwerk C angemeldet). Da man unter TOS nicht in der Lage ist, die gesamte Platte anzusprechen, muß der Programmierer einen anderen Weg, nämlich den, die Platte direkt anzusteuern, gehen.

Anzusprechen ist eine Festplatte am Atari ST nur über den DMA-Chip. Er stellt die Hardware-Schnittstelle zu den externen Massenspeichern wie Diskettenlaufwerken und Festplatten dar und entlastet den Prozessor, indem er unabhängig von diesem arbeitet. Man erreicht den DMA über folgende Hardware-Register (Achtung: nur im Supervisor-Modus ansprechbar !!):

FF8604 (diskctl): Allgemeines Eingabe-/Ausgaberegister

FF8606 (fifo): Das Modusregister. Hiermit wird der DMA-Chip gesteuert.

FF8609 (dmahigh): DMA-Adresse HI-Byte der zu schreibenden/lesenden Daten.

FF860B (dmamid): DMA-Adresse MID-Byte

FF860D (dmalow): DMA-Adresse LO-Byte Die für den Harddisk-Betrieb wichtigen Bits des Steuerregisters des DMA- Chips sind diese:

Bit 1: Adreßleitung A0, wenn gesetzt, wird der Controller angesprochen

Bit 3: 0 = Zugriff auf den FDC (Floppy-Disk-Controller)
1 = Zugriff auf den HDC (Harddisk-Controller)

Bit 4: 1 = Zugriff auf den DMA-Sektor-Zähler.

Bit 7: 0 = Zugriff auf den HDC
1 = Zugriff auf den FDC

Bit 8: 0 = DMA: in den Speicher einlesen
1 = DMA: aus dem Speicher schreiben

Wie muß das Programm vorgehen, um den Bootsektor von der Festplatte lesen zu können ?

Zunächst wird über Drvmap() geprüft, ob überhaupt Partitionen der Festplatte als zusätzliche Stationen angemeldet sind. Dies ist eine reine Sicherheitsmaßnahme, denn wäre die Harddisk nicht in Betrieb, würde der Computer beim Zugriff auf den nicht vorhandenen HDC abstürzen.

Wie schon gesagt, haben alle Zugriffe auf den DMA-Chip im Supervisor-Modus zu erfolgen. Das Programm schaltet den Prozessor mittels super_on() um. Auch muß ein jetzt unerwünschter Eingriff des Betriebssystems in den Datenaustausch mit dem DMA-Chip und dem Festplatten-Controller unterbunden werden. Dazu wird die Systemvariable FLOCK auf einen Wert ungleich 0 gesetzt. Siehe dazu auch die Routine hd_read().

Mit set_buf() wird dem DMA-Chip die Adresse des Zielpuffers für den gelesenen Sektor mitgeteilt. Die Routine seek_block überträgt die Daten wie Sektornummer und Anzahl zum Plattencontroller. Er benötigt für den Lesezugriff eine Folge von fünf Bytes: (siehe Routine seek_block)

1.Byte: Befehl (Lesen = $08), in die obersten 3 Bits ist die Controllernummer der Platte einzusetzen.

2.Byte: Sektor-Adresse, High-Byte. In die oberen 3 Bits ist die Nummer des Laufwerkes einzusetzen.

3.Byte: Sektor-Adresse, Mid-Byte (ohne Laufwerksnummer).

4.Byte: Sektor-Adresse, Low-Byte.

5.Byte: Anzahl der zu lesenden Sektoren.

6.Byte: Wird hier nicht benutzt.

Dieses Protokoll entspricht übrigens dem des SCSI-Bus-Systems. Zusätzlich muß der DMA-Chip bei jedem zu übertragenden Byte auf $8A gesetzt werden, damit die Daten auch wirklich an den richtigen Controller gelangen und nicht aus Versehen beim FDC landen. Auch ist zu sagen, daß die Sektoren einer Festplatte fortlaufend durchnumeriert sind, also bei der Atari-Festplatte von #0 bis #41615.

Zwischen den Übertragungen der einzelnen Bytes muß dem Controller etwas Zeit gelassen werden, um den Befehl zu verarbeiten. Wie auch der Floppycontroller meldet er, wenn er wieder empfangsbereit ist, indem er das Bit 5 im MFP-Register GPIP (Adresse FFFA01) auf Null setzt (siehe Routine Timer). Tut er das innerhalb von 1/20 Sekunden jedoch nicht, so darf man davon ausgehen, daß bei ihm irgendetwas schiefgelaufen ist. Die Routine seek_block wird dann mit einer Fehlermeldung verlassen.

Hat der HDC jedoch alles geschluckt, was ihm der DMA-Chip hinübergeschickt hat, so wird er nun den erhaltenen Befehl ausführen, den Bootsektor von der Platte lesen und zum DMA-Chip senden. Zuerst muß aber auch der DMA-Chip noch darauf eingestellt werden, die Daten zu empfangen und an die gewünschte Adresse zu schreiben (siehe Routine dma_read()). Dazu muß einmal kurz mit der Schreib-Leseleitung in Bit 8 des Modus-Registers 'geklappert' werden, damit der Status des DMA-Chips gelöscht wird. Nun noch die gewünschte Anzahl der zu übertragenden Sektoren in FF8604 eintragen und die Übertragung kann beginnen.

Nachdem der DMA-Chip die Daten, in diesem Falle den Bootsektor, in den Speicher geschrieben hat, überprüft die Routine status() noch, ob auch alles in Ordnung ist. Ist der Sektorzähler (erreichbar in *diskctl) nicht gleich Null, so konnten die Daten nicht korrekt übertragen werden und die Routine gibt Null als Fehlercode zurück.

```
HARDDISK-INFORMATION V2.0  Von Ulrich Neumann
--------------------------------------------------------------------------------
Gesamte Größe  :    43996 Sektoren =    21.482 MBytes
Wahre Größe    :    43996 Sektoren =    21.482 MBytes
Davon belegt   :    9.724 MBytes   =    45.26  %
Freier Platz   :   11.759 MBytes
Keine defekten Sektoren vorhanden.
--------------------------------------------------------------------------------
Partition 1 : GEM und bootbar.
Größe :    10880 Sektoren =    5.313 MBytes
Belegt:    1.089 MBytes   =    20.50  %      Frei :    4.224 Mbytes
--------------------------------------------------------------------------------
Partition 2 : GEM
Größe :    10880 Sektoren =    5.313 MBytes
Belegt:    3.681 MBytes   =    69.28  %      Frei :    1.632 Mbytes
--------------------------------------------------------------------------------
Partition 3 : GEM
Größe :    22144 Sektoren =   10.812 MBytes
Belegt:    4.909 MBytes   =    45.40  %      Frei :    5.903 Mbytes
--------------------------------------------------------------------------------
Partition 4 : ?EM
Größe :        0 Sektoren =    0.000 MBytes
Belegt:    0.000 MBytes   =    0.00   %      Frei :    0.000 Mbytes
```

Bild 1: So meldet sich die Harddisk-Information

Zu guter Letzt muß der DMA-Chip wieder in den Urzustand versetzt werden, indem man den FDC selektiert (Bit 7 setzen, alle anderen löschen). Außerdem sollte die Sperre für den Zugriff des TOS auf den DMA-Chip wieder aufgehoben werden, und zwar mit *flock = OFF. Da ab jetzt keine Zugriffe mehr auf geschützte Speicherbereiche erfolgen, kann der Supervisormodus mit super_off() verlassen werden.

Auf diese Weise kommt das Programm also an die Information im Bootsektor heran und kann diese nunmehr auswerten:

Es geht alle vier Partition-Einträge durch. Sollte eine der Partitionen den Id-Code 'GEM' haben, so handelt es sich hierbei um eine GEMDOS-Partition und man kann über Dfree() den freien Platz auf ihr erhalten (siehe Routine do_info()). Aus diesen Werten wird dann der noch freie Platz auf der gesamten Platte summiert.

Die Routine disk_info() gibt zuerst die Informationen über die Festplatte als Gesamtes aus und die Routine partitions() übernimmt die Ausgabe der Werte für die einzelnen Partitionen. Mit einem Tastendruck kehrt man schließlich ins Desktop zurück.

Noch ein Wort zur Routine hd_read(): sie ist so programmiert, daß man auch beliebige andere Sektoren von der Harddisk lesen kann, auch mehrere. Da die SH205 die Möglichkeit bietet, mehrere Systeme hintereinanderzuschalten, kann hd_read() auch alle weiteren Controller und Laufwerke ansprechen, indem man deren Nummern beim Aufruf übergibt (siehe Listing). Zwei Beispiele hierzu stellen die Routinen hd_write() und hd_park() dar.

Hd_write() in Verbindung mit dma_write() kann bis zu 255 Sektoren aus dem Speicher an beliebige Stellen auf der Festplatte schreiben. Hier wird der Lese-Befehl $0A an den Festplatten-Controller gesendet. Aber Vorsicht: diese Funktion ist in ihrer Wirkung in etwa vergleichbar mit einem Skalpell oder einer Motorsäge, besonders weil die Festplatte im Gegensatz zur 3.5"-Diskette kein Schreibschutzloch hat. Also gilt: absolute Vorsicht beim Einsatz. Genauso würde ich den Befehl $04 "Format Drive" nur dem HDX-Programm überlassen, wenn Sie ihre Daten noch einmal lebend wiedersehen wollen!

Ein weiteres, wesentlich harmloseres Beispiel, das eher zur Datensicherheit beiträgt, ist die Funktion hd_park(): Sie erfüllt die gleiche Aufgabe wie das SHIP-Programm von Atari - fährt also die Leseköpfe in die Parkposition - hat aber den Vorteil, daß sie auch in Accessories eingebaut werden kann. Nach dem Aufruf der Funktion sollte der Lesekopf nicht mehr angesprochen werden. Erst nach erneutem Hochfahren der Festplatte (also aus- und wieder einschalten) macht er wieder, was der Computer will.

Diese Routinen kann sich jeder nach eigenem Gutdünken in seine Programme einbauen, ich denke da z.B. an einen Harddisk-Monitor.
Das Programm wurde in Megamax C geschrieben, sollte aber auch auf den 'Kollegen' wie DRI C oder Mark Williams C laufen. Eventuell muß man dann die Datentypen an den Compiler anpassen.
Ich hoffe, daß dieses Programm dem geneigten Leser etwas Einsicht in den Betrieb mit der Atari-Festplatte gewährt hat. Sollten noch Fragen auftauchen, so stehe Ich gerne zur Verfügung. Ich übernehme aber keine Verantwortung für 'abgeschossene' Bootsektoren oder ähnliche Schwierigkeiten, die z.B. durch Fehler beim Abtippen entstehen können.

Ulrich Meumann

```
  1: /**************************************************/
  2: /*                                                */
  3: /*     HARDDISK - INFORMATION  TOS-Version 2.0    */
  4: /*                                                */
  5: /* Programm zum Ausgeben von Informationen über   */
  6: /* die Belegung von SH204/205 und Festplatten mit */
  7: /* kompatiblen Controllern (ST506/412/SCSI)       */
  8: /**************************************************/
  9: /*            Geschrieben in Megamax C von        */
 10: /*                                                */
 11: /*                 Ulrich Meumann                 */
 12: /*                 Lehnerweg 2, 7900 Ulm          */
 13: /*                 Tel. 0731/382100               */
 14: /**************************************************/
 15:
 16: #include <osbind.h>
 17:
 18: #define ON  -1
 19: #define OFF 0
 20:
 21: #define READ 0x08  /* Controller: Von Platte lesen */
 22:
 23: #define LANG   400
 24: #define KURZ   10
 25:
 26: struct partition { /* Partition-Info im Bootblock */
 27:                  char  p_flag;  /* Flag, <>0, wenn
                                       Part. gültig */
 28:                  char  p_id1;   /* Drei ID-Bytes */
 29:                  char  p_id2;
 30:                  char  p_id3;
 31:                  long  p_start; /* Startsektor der
                                       Partition */
 32:                  long  p_size;  /* Länge in
                                       Sektoren */
 33: };
 34:
 35: struct bootblock {
 36:                  char  nutzlos[0x1c0];
                     /* Je nach Autobooter versch. */
 37:                  char  hi_spt;
                     /* Sektoren pro Spur (SH205: 17)*/
 38:                  long  hd_size;
                     /* Länge der Platte in Sektoren */
 39:                  struct partition p_table[4];
                     /* Vier Partition-Einträge */
 40:                  long  bsl_start;
                     /* Anfang der defekten Sektoren */
 41:                  long  bsl_count;
                     /* Anzahl der defekten Sektoren */
 42:                  int   checksum;
                     /* Checksumme für den Autoboot */
 43: };
 44:
 45: char puffer[512];  /* Hier kommt der Bootblock rein*/
 46:
 47: struct bootblock *boot = puffer;
 48:
 49: char  *gpip = 0xFFFA01L; /* I/O-Register im MFP */
 50:
 51: int   *diskctl_w = 0xFF8604L; /* HDC-Register in der
                                    DMA */
 52: long  *diskctl_l = 0xFF8604L; /* Das Gleiche in
                                    long */
 53: int   *fifo      = 0xFF8606L; /* Steuerregister der
                                    DMA */
 54: char  *dmahigh   = 0xFF8609L; /* DMA-Adresse,
                                    High-Byte */
 55: char  *dmamid    = 0xFF860BL; /* Mid-Byte */
 56: char  *dmalow    = 0xFF860DL; /* Low-Byte */
 57: int   *hz_200    = 0x4BAL;    /* Der 200Hz-
                                    Systemzähler */
 58: int   *flock     = 0x43EL;    /* Sperren der
                                    Floppy-VBL's */
 59:
 60: long save_ssp;     /* Supervisor-Stackpointer */
 61:
 62:
 63:
 64: main()
 65: {
 66:    if (get_boot())
 67:       do_info();
 68:    else
 69:       Cconws("Leider kann keine Information
                  ausgegeben werden !!\n\r");
 70:    Cconin();
 71: }
 72:
 73: /*********************************/
 74: /*  Bootsektor von Harddisk lesen  */
 75: /*********************************/
 76:
 77: int get_boot()
 78: {
 79:    if(Drvmap() & 60) /* Sind über Stationen
                             angemeldet ?*/
 80:       return(hd_read(0L,1,puffer,0,0));
 81:    else
 82:       return(0);
 83: }
 84:
 85: /**************************************************/
 86: /* >anzahl< Sektoren ab >sektor< vom Laufw. >drv< */
 87: /*          mit dem  Controller >crtl< in         */
 88: /*              den Puffer >buf< laden            */
 89: /**************************************************/
 90: int hd_read(sektor, anzahl, buf, drv, crtl)
 91: long sektor, buf;
 92: int anzahl, drv, crtl;
 93: {
 94:    int ok;
 95:    super_on();    /* Supervisor-Modus einschalten */
 96:    *flock = ON;   /* Floppy-VBL sperren */
 97:    set_buf(buf); /* Adresse des Puffers eintragen */
 98:    ok = seek_block(READ, sektor, anzahl, drv, crtl);
                           /* Sektoren suchen */
 99:    if (ok)
100:       dma_read(anzahl); /* DMA auf Lesen
                                programmieren */
101:    ok = status(); /* Hat's auch geklappt ??? */
102:    *fifo = 0x80;  /* DMA wieder in Normalzustand */
103:    *flock = OFF;  /* Floppy-VBl wieder freigeben */
104:    super_off();   /* Zurück in den User-Modus */
105:    return(ok);    /* Das war's!! */
106: }
107:
108: set_buf(buf)    /* Pufferadresse in DMA eintragen */
109: long buf;
110: {
111:    *dmalow = (char)(buf & 0xFF);
112:    *dmamid = (char)((buf >> 8) & 0xFF);
113:    *dmahigh= (char)((buf >> 16) & 0xFF);
114:
115: }
116:
117: /*************************************/
118: /*  Sektor(en) auf der Platte anfahren  */
119: /*************************************/
120:
121: int seek_block(befehl, sektor, anzahl,
                   drive, controller);
122: int befehl, anzahl, drive, controller;
123: long sektor;
124: {
125:    long kommando;
126:    int fehler;
127:
128:    *fifo = 0x88;
129:
130:    kommando = ((long)controller << 21) |
                  ((long)befehl << 16) | 0x8a;
131:    *diskctl_l = kommando; /* 1.Byte an den
                                 Controller schicken*/
132:    if ((fehler = timer(KURZ)) == 0) /* 1/20 Sekunde
                                           warten, bis */
133:       return(fehler);               /* der Controller
                                           bereit ist */
134:
135:    *diskctl_l = ((long)drive << 21) |
                     (sektor & 0xFF0000) | 0x8A;
```

```
136:     if ((fehler = timer(KURZ)) == 0) /* 2.Byte senden
                                           und warten */
137:        return(fehler);
138:
139:     *diskctl_l = (sektor & 0xFF00) << 8 | 0x8A;
                                   /* 3.Byte senden */
140:     if ((fehler = timer(KURZ)) == 0)
141:        return(fehler);
142:
143:     *diskctl_l = (sektor & 0xFF) << 16 | 0x8A;
                                   /* 4.Byte senden */
144:     if ((fehler = timer(KURZ)) == 0)
145:        return(fehler);
146:
147:     *diskctl_l = ((long)anzahl & 0xFF) << 16 | 0x8A;
                                   /* Anzahl senden */
148:     return(timer(KURZ));
149: }
150:
151: /**************************************************/
152: /*  Auf die Bereit-Meldung des Controllers warten, */
153: /*  aber höchstens zeit/20 Sek.                    */
154: /**************************************************/
155: timer(zeit)
156: int zeit;
157: {
158:     int wert;
159:
160:     wert = *hz_200 + zeit;
161:     while (*hz_200 != wert)
162:        if ((*gpip & 0x20) == 0) /* Hat der Controller
                                      sich gemeldet ?*/
163:           return(-1);
164:     return(0);           /* Zeit abgelaufen: Fehler! */
165: }
166:
167: /**************************************************/
168: /* DMA auf Lesen von >anzahl< Blocks programmieren.*/
169: /**************************************************/
170:
171: dma_read(anzahl)
172: int anzahl;
173: {
174:     *fifo = 0x98; /* Mit der Schreib-Lese-Leitung
                         klappern..*/
175:     *fifo = 0x198;
176:     *fifo = 0x98;
177:     *diskctl_w = anzahl; /* Anzahl eintragen...*/
178:     *fifo = 0x8A;
179:     *diskctl_l = 0L;
180: }
181:
182: /**************************************/
183: /* Statusmeldung vom Controller holen */
184: /**************************************/
185:
186: int status()
187: {
188:     int fehler;
189:
190:     if ((fehler = timer(LANG)) == 0)
                          /* Zwei Sekunden warten */
191:        return(0);       /* Fehler, weil Zeit alle */
192:     *fifo = 0x8A;
193:     fehler = *diskctl_w & 0xFF;
194:     return(!fehler);
195: }
196:
197: super_on()          /* Supervisor-Modus einschalten */
198: {
199:     save_ssp = Super(0L);
200: }
201:
202: super_off()         /* Zurück in den User-Modus */
203: {
204:     Super(save_ssp);
205: }
206:
207: /**********************************************/
208: /*  Information über die Festplatte ausgeben  */
209: /**********************************************/
210:
211: long p_frei[4], hd_frei, p_lang[4];
212: char p_str[4];
213:
214: do_info()
215: {
216:     long pbuff[4];
217:     int i;
218:
219:     hd_frei = 0;
220:
221:     Cconws("                    \033p HARDDISK-INFORMATION
               V2.0  Von Ulrich Meumann \033q\n\r");
222:     Cconws("-------------------------------------------
               ---------------------------------\r");
223:
224:     for(i=0;i<4;i++)
225:     {
226:        if(boot->p_table[i].p_flag &&
              boot->p_table[i].p_id1=='G' &&
              boot->p_table[i].p_id2=='E')
227:        {
228:           Dfree(pbuff,i+3); /* Freier Platz auf der
                                    Partition */
229:           p_frei[i] = pbuff[0]*1024;
230:        }
231:        else
232:           p_frei[i] = 0;
233:        p_lang[i] = boot->p_table[i].p_size;
           /* Länge der Partition in Sektoren */
234:        hd_frei += p_frei[i];
           /* Freier Platz auf der ganzen Platte */
235:     }
236:
237:     disk_info();  /* Information über die
                          Gesamtbelegung */
238:     partitions(); /* Informationen z.d.Partitions */
239: }
240:
241: /*******************************************/
242: /* Informationen zur gesamten Platte ausgeben */
243: /*******************************************/
244:
245: disk_info()
246: {
247:     long ws;
248:
249:     printf(" Gesamte Größe  : %8ld Sektoren",
            boot->hd_size);
250:     printf(" = %8.3f MBytes\n",
            ((float)boot->hd_size*512)/1024/1024);
251:
252:     ws = (boot->hd_size - boot->bsl_count);
253:
254:     printf(" Wahre Größe    : %8ld Sektoren",ws);
255:     printf(" = %8.3f MBytes\n",
            ((float)ws*512)/1024/1024);
256:
257:     printf(" Davon belegt   : %8.3f MBytes",
          ((float)ws*512-hd_frei)/1024/1024);
258:     printf("   = %7.2f  %%\n",
          ((float)(ws*512-hd_frei)/(float)(ws*512)*100));
259:
260:     printf(" Freier Platz   : %8.3f MBytes\n",
          ((float)hd_frei)/1024/1024);
261:
262:     if (boot->bsl_count)
263:     {
264:        printf(" Defekte Sektoren: %5ld",
              boot->bsl_count);
265:        printf(" = %8.1fKBytes\n",
              ((float)boot->bsl_count*512)/1024);
266:     }
267:     else
268:        printf(" Keine defekten Sektoren
                 vorhanden.\n");
269: }
270:
271: /**************************************************/
272: /*Informationen z.d. einzelnen Partitionen ausgeben*/
273: /**************************************************/
274:
275: partitions()
276: {
277:     int i;
278:     long belegt;
279:     char p_code[4];
280:
281:     for(i=0;i<4;i++)
282:     {
283:        Cconws("-------------------------------------
              --------------------------------------\r");
284:        p_code[0]=boot->p_table[i].p_id1;
285:        p_code[1]=boot->p_table[i].p_id2;
286:        p_code[2]=boot->p_table[i].p_id3;
287:        p_code[3]='\0';
288:        printf(" Partition %d : ",i+1);
289:        if (boot->p_table[i].p_flag)
290:           printf("%s ",p_code);
291:        else
292:           printf("Nicht benutzt.\n");
293:        if (boot->p_table[i].p_flag & 0x80)
294:           printf("und bootbar.\n");
295:        else
```

```
296:            printf("\n");
297:         if (boot->p_table[i].p_flag)
298:         {
299:            printf(" Größe : %8ld Sektoren",p_lang[i]);
300:            printf(" = %8.3f MBytes\n",
                   ((float)p_lang[i]*512)/1024/1024);
301:            belegt = p_lang[i]*512-p_frei[i];
302:            printf(" Belegt: %8.3f MBytes",
                   ((float)belegt)/1024/1024);
303:            printf("   = %7.2f  %%",
                 ((float)belegt/(float)(p_lang[i]*512))*100);
304:            printf("      Frei : %8.3f Mbytes\n",
                   (float)p_frei[i]/1024/1024);
305:         }
306:      }
307:   }
308:
309:   /**********************************************/
310:   /* ZUSÄTZLICHE FUNKTIONEN:SCHREIBEN UND PARKEN */
311:   /**********************************************/
312:
313:   #define WRITE 0x0A
314:
315:   /************************************************/
316:   /* >anzahl< Sektoren ab >sektor< vom Puffer >buf<  */
317:   /* auf das Laufwerk >drv< mit dem Controller        */
318:   /* >crtl< schreiben.                                */
319:   /************************************************/
320:   int hd_write(sektor, anzahl, buf, drv, crtl)
321:   long sektor, buf;
322:   int anzahl, drv, crtl;
323:   {
324:      int ok;
325:      super_on();     /* Supervisor-Modus einschalten */
326:      *flock = ON;    /* Floppy-VBL sperren */
327:      set_buf(buf);   /* Adresse des Puffers eintragen */
328:      ok = seek_block(WRITE, sektor, anzahl, drv, crtl);
                                      /* Sektoren suchen */
329:      if (ok)
330:         dma_write(anzahl); /* DMA auf Schreiben
                                  programmieren */
331:      ok = wr_status();     /* Hat's auch geklappt ??? */
332:      *fifo = 0x80;    /* DMA wieder i.d. Normalzustand */
333:      *flock = OFF;    /* Floppy-VBl wieder freigeben */
334:      super_off();     /* Zurück in den User-Modus */
335:      return(ok);      /* Das war's!! */
336:   }
337:
338:   /************************************************/
339:   /*   DMA auf Schreiben von >anzahl< Blocks progr.  */
340:   /************************************************/
341:
342:   dma_write(anzahl)
343:   int anzahl;
344:   {
345:      *fifo = 0x98; /* Mit der Schreib-Lese-Leitung
                         klappern..*/
346:      *fifo = 0x198;
347:      *diskctl_w = anzahl; /* Anzahl eintragen...*/
348:      *fifo = 0x18A;
349:      *diskctl_l = 0x100L;
350:   }
351:
352:   /***************************************/
353:   /* Statusmeldung vom Controller holen */
354:   /***************************************/
355:
356:   int wr_status()
357:   {
358:      int fehler;
359:
360:      if ((fehler = timer(LANG)) == 0) /* Zwei Sekunden
                                            warten  */
361:         return(0);          /* Fehler, weil Zeit alle */
362:      *fifo = 0x18A;
363:      fehler = *diskctl_w & 0xFF;
364:      return(!fehler);
365:   }
366:
367:   #define PARK 0x1B
368:
369:   /**********************************************/
370:   /* Köpfe der Platte >drv< mit HDC >crtl< in */
371:   /* die Parkposition fahren...                  */
372:   /**********************************************/
373:   int hd_park(drv,crtl)
374:   int drv, crtl;
375:   {
376:      int ok;
377:      super_on();     /* Supervisor-Modus einschalten */
378:      *flock = ON;    /* Floppy-VBL sperren */
379:      ok = seek_block(PARK, 0L, 0, drv, crtl);
                                     /* Sektoren suchen */
380:      *diskctl_l = (long)0x0a; /* Noch ein Byte
                                    senden..*/
381:      ok = timer(KURZ);  /* Und nochmal warten... */
382:      ok = status();     /* Hat's auch geklappt ??? */
383:      *fifo = 0x80;      /* DMA wieder in den
                               Normalzustand */
384:      *flock = OFF;      /* Floppy-VBl wieder
                               freigeben */
385:      super_off();       /* Zurück in den User-Modus */
386:      return(ok);        /* Das war's!! */
387:   }
```

ENDE

HARDCOPIES VON EINZELNEN FENSTERN

Wer einmal auf dem MacIntosh gearbeitet hat, kennt die praktische Möglichkeit, nur den Inhalt eines Fenster auf dem Drucker auszugeben. Damit auch die ATARI ST-Benutzer ihr Papier nicht mehr nur überwiegend mit dem Graumuster des Desktops füllen müssen, wurde das Accessory PRINTTOP geschrieben.

Mit ihm kann das jeweils aktive Fenster an den Drucker als Hardcopy geschickt werden. Dazu brauchen Sie nur den Eintrag "Print Window" im Deskmenü auszuwählen und zu bestätigen.

Das Programm baut dazu einen Bildschirm auf, der nur im Speicher vorhanden ist, aber nicht angezeigt wird. In ihn wird der Inhalt des obersten Fensters kopiert und dann die normale Hardcopy-Routine des Betriebssystems aufgerufen.

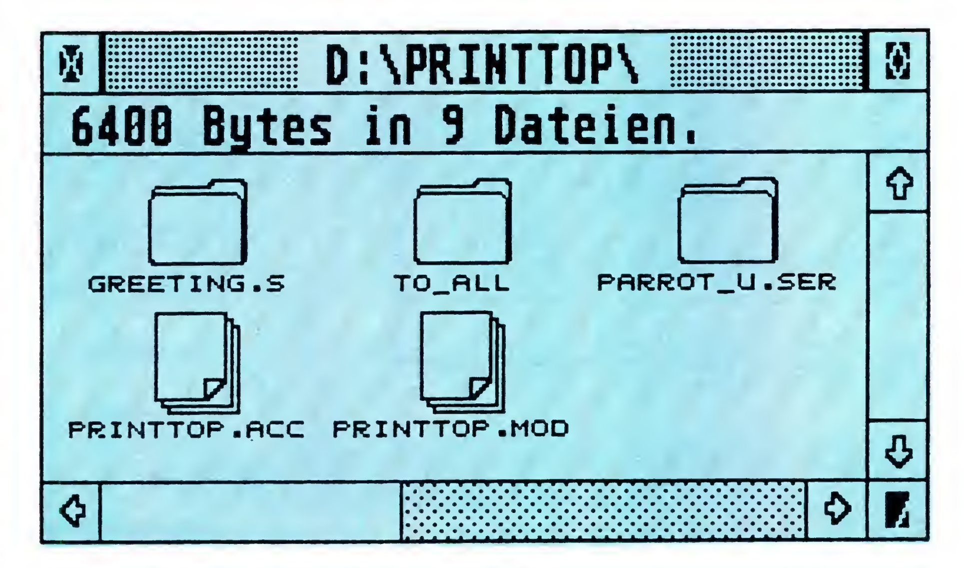
Aber zunächst müssen natürlich die Kennzahl sowie Größe und Lage des Fensters festgestellt werden. Das geschieht in den (nur zur Orientierung numerierten) Zeilen 66 bis 70. Graf_Handle liefert die Kennzahl der VDI-Arbeitsstation, auf der das AES zeichnet. Der erste WindowGet-Aufruf ergibt die Kennzahl des obersten Fensters, dessen Dimensionen im zweiten WindowGet-Call abgefragt werden.

Falls ein Fenster vorhanden war, wird es zunächst zurechtgestutzt, wenn es teilweise außerhalb des Bildschirms lag. Um den privaten Bildschirm zu löschen, teilt das Programm dem Betriebssystem zunächst mit, daß er ab jetzt an einer anderen Speicheradresse liegt. Die VT52-Sequenz ESC E arbeitet dann wie gewünscht. Mit einem Rastercopy kann nun der Fensterinhalt in den Privatbildschirm kopiert werden. Das Ausdrucken geschieht dann einfach mit einem ScreenDump-Aufruf, der den logischen und nicht den sichtbaren Bildschirmspeicher ausgibt. Schließlich muß natürlich noch die Bildschirmadresse auf den urspünglichen Wert gebracht werden.

Bei der Einrichtung des privaten Bildschirms im Speicher wird zunächst ein entsprechender Block von GEMDOS angefordert und dann die Adresse so aufgerundet, daß der Bildschirm an einer 512-Byte Grenze beginnt (Zeilen 41 bis 59). Das muß geschehen, da die Systemroutinen mit dem Bildschirm nur unter dieser Voraussetzung arbeiten können.

Der Rest des Programms besteht aus den üblichen Befehlen bei der Installation einen Accessories und Fehlerbehandlungen. Falls für den privaten Bildschirm nicht genügend Speicher frei war, sollten Sie andere Accessories entfernen oder die RAM-Disk verkleinern. Wenn Ihr System aber keine 32 KByte nach dem Booten mehr frei hat, können Sie sowieso kaum etwas starten, so daß dieser Fall praktisch nie auftreten wird.

Da die Standard-Hardcopy benutzt wird, ist keine Druckeranpassung notwendig. Wenn Sie z.B. einen 24-Nadel-Drucker oder einen anderen nicht EPSON-kompatiblen besitzen, kann PRINTTOP dann arbeiten, wenn ein spezieller Hardcopytreiber installiert ist. Für die P6/P2200-Drucker von NEC ist ein solches Programm als public-domain verfügbar, wie auch für andere.



Ausdruck eines Fensters mit dem PRINTTOP-Accessory

Wenn es Sie stört, daß der Fensterrahmen mitausgedruckt wird, brauchen Sie nur in Zeile 70 "CurrXYWH" und "WorkXYWH" auszutauschen. Das AES liefert dann die Dimensionen der inneren Fensterfläche beim Window-Get-Aufruf.

Im Bild sehen Sie eine Fensterhardcopy, die mit PRINTTOP erstellt wurde. Dabei mußten keine störenden grauen Flächen vom Desktop-Hintergrund mehr entfernt werden.

Robert Tolksdorf

```
  1:  MODULE PrintTopWindow;
  2:
  3:  (* Robert Tolksdorf 5.2.88 *)
  4:
  5:  (*$A+*) (* Optimieren *)
  6:
  7:  FROM SYSTEM       IMPORT ADR, ADDRESS ;
  8:  FROM GEMAESbase   IMPORT AccessoryOpen, Top, CurrXYWH,
  9:                           MouseOff, MouseOn, AESCallResult ;
 10:  FROM AESApplications  IMPORT ApplInitialise;
 11:  FROM AESForms     IMPORT FormAlert;
 12:  FROM AESGraphics  IMPORT GrafHandle, GrafMouse ;
 13:  FROM AESMenus     IMPORT MenuRegister;
 14:  FROM AESEvents    IMPORT EventMessage;
 15:  FROM AESWindows   IMPORT WindowGet;
 16:  FROM GEMVDIbase   IMPORT BigPxyArrayType;
 17:  FROM VDIRasters   IMPORT MFDBType, CopyRasterOpaque ;
 18:  FROM GEMDOS       IMPORT Alloc, ConOut ;
 19:  FROM XBIOS        IMPORT ScreenLogicalBase,
 20:                               SetScreenBase, ScreenDump ;
 21:
 22:  CONST Title       = '  Print Window';
 23:        NoMemAlert  = '[1][Nicht genug| Speicher frei]
                         [OK]';
 24:        RebootAlert = 'Sorgen Sie für mehr Speicher|und
                         booten Sie neu][OK]';
 25:        ReallyAlert = '[3][Fenster|ausdrucken ?]
                         [OK|CANCEL]';
 26:        NoWindAlert = '[1][Kein Fenster|vorhanden !]
                         [OK]';
 27:  VAR
 28:    Msg                 : ARRAY [0..16] OF INTEGER;
 29:    SHandle, tophandle,
 30:    applID, menuID,
 31:    x, y, w, h          : INTEGER;
 32:    mymemory            : ADDRESS;
 33:    ScreenMFDB, MyMFDB  : MFDBType;
 34:    pxyarray            : BigPxyArrayType;
 35:
 36:  BEGIN
 37:  (* Application initialisieren u. Accessory anmelden *)
 38:    applID := ApplInitialise();
 39:    menuID := MenuRegister(applID,Title);
 40:    (* Speicher für Privatbildschirm anfordern *)
 41:    Alloc(32511,mymemory);
 42:    IF mymemory<=0 THEN
 43:      (* Kein Speicher frei ... *)
 44:      x:=FormAlert(1,NoMemAlert)+
            FormAlert(1,RebootAlert);
 45:      LOOP EventMessage(ADR(Msg)) END
 46:    ELSE
 47:      (* MFDB's setzen *)
 48:      WITH ScreenMFDB DO
 49:        pointer:=ScreenLogicalBase();
 50:        width:=639;   (* mono-Bildschirm !! *)
 51:        height:=399;
 52:        widthW:=40;
 53:        format:=1;
 54:        planes:=1
 55:      END;
 56:      MyMFDB:=ScreenMFDB;
 57:      (* Privatbildschirm an 1/2 KByte-Grenze setzen *)
 58:      MyMFDB.pointer:=
 59:        ADDRESS(LONGCARD(mymemory)+512-(LONGCARD
              (mymemory) MOD 512));
 60:      LOOP
 61:        (* Auf Mitteilung warten *)
 62:        EventMessage(ADR(Msg));
 63:        (* Fenster ausdrucken ??? *)
 64:        IF (Msg[0]=AccessoryOpen) AND (FormAlert(1,
             ReallyAlert)=1) THEN
 65:          (* AES VDI-Handle holen *)
 66:          SHandle:=GrafHandle(x,x,x,x);
 67:          (* Handle des obersten Fensters in tophandle
             holen *)
 68:          WindowGet(SHandle,Top,tophandle,x,x,x);
 69:          (* Größe des obersten Fensters holen *)
 70:          WindowGet(tophandle,CurrXYWH,x,y,w,h);
 71:          IF AESCallResult>0 THEN
 72:          (* Clippen *)
 73:            IF x+w>INTEGER(ScreenMFDB.width) THEN
 74:              w:=INTEGER(ScreenMFDB.width)-x+1;
 75:            END;
 76:            IF y+h>INTEGER(ScreenMFDB.height) THEN
 77:              h:=INTEGER(ScreenMFDB.height)-y+1;
 78:            END;
 79:          (* Maus ausschalten *)
 80:            GrafMouse(MouseOff,NIL);
 81:          (* logisch auf Privatbildschirm umschalten *)
 82:            SetScreenBase(MyMFDB.pointer,NIL,-1);
 83:          (* Privatbildschirm löschen *)
 84:            ConOut(CHR(27)); ConOut('E');
 85:          (* Fenster in Privatbildschirm kopieren *)
 86:            pxyarray[0]:=x;        pxyarray[1]:=y;
 87:            pxyarray[2]:=x+w-1;    pxyarray[3]:=y+h-1;
 88:            pxyarray[4]:=0;        pxyarray[5]:=0;
 89:            pxyarray[6]:=w-1;      pxyarray[7]:=h-1;
 90:            CopyRasterOpaque(SHandle,3,pxyarray,
                  ADR(ScreenMFDB),ADR(MyMFDB));
 91:          (* Dumpen *)
 92:            ScreenDump;
 93:          (* logisch wieder auf eigentlichen Bildschirm
                 umschalten *)
 94:            SetScreenBase(ScreenMFDB.pointer,NIL,-1);
 95:          (* Maus einschalten *)
 96:            GrafMouse(MouseOn,NIL);
 97:          ELSE
 98:            x:=FormAlert(1,NoWindAlert)
 99:          END;
100:        END
101:      END
102:    END
103:  END PrintTopWindow.
```

ENDE

NO_CLS

ein Patch für GFA-BASIC

In compiliertem GFA-BASIC wird beim Laden des Programm-Codes grundsätzlich der Bildschirm gelöscht. Dafür sorgt ein 'Initialisierungs-String', der in dem Compilat ab dem 30. Byte enthalten ist. Nun ist es aber besonders bei Programmen, die mit einer GEM-Umgebung arbeiten nicht wünschenswert, daß der Bildschirm gelöscht wird, da es stümperhaft aussieht, wenn der Hintergrund erst gelöscht und dann wieder aufgebaut wird. Mein kleines Programm ändert deshalb ein compiliertes Programm so ab, daß der Bildschirm nach dem Laden erhalten bleibt.

Die Arbeitsweise des Programmes

In einer Fileselektorbox wird der Name eines (compilierten GFA-BASIC-) Programmes eingegeben. Dieses Programm wird dann 'geöffnet' und der Dateipointer mittels des SEEK-Befehles auf das 30. Byte gesetzt. An dieser Stelle beginnt nun normalerweise der String, der den Bildschirm mit Hilfe einer Escape-Sequenz (wie 'Print Chr$(27); Chr$(69)' oder auch 'Cls') löscht (siehe Tabelle). Als Kennung für das Ende dieses Strings gilt das ASCII-Zeichen mit der Nummer 0. Also müssen wir nur das erste Byte dieses Strings 'Auf Null setzen', damit auf dem Schirm nichts passiert.

Das erledigt für uns der Befehl OUT #1,0. Dieser Befehl schreibt das Byte Null an die Stelle, auf die der Dateipointer zeigt. Da wir den Dateipointer bereits auf das 30. Byte gesetzt haben, wird es folglich mit Null überschrieben.

Wenn Sie das 'behandelte' Programm dann das nächste Mal laden, werden Sie feststellen, daß Ihnen der Bildschirm erhalten bleibt. Die Menüzeile, in der während des Ladevorganges der Name des Programmes steht, ist dann leer. Darunter bleibt der Schirm im grauen Farbton des GEM-Desktops.

Dieses Verändern des Compilats wird in einer READ_ME-Datei auf der GFA-BASIC-Compiler-Diskette vorgeschlagen. Somit brauchen Sie keine Angst zu haben, daß ein solches Vorgehen zu Datenverlusten o.ä. führen könnte.

Obwohl sich der Zeitaufwand für das Behandeln eines Programmes in engem Rahmen hält, wäre es doch sehr wünschenswert, direkt in den Compiler-Optionen einstellen zu können, ob der Bildschirm gelöscht oder 'normal' übergeben wird. Auch dies wäre eine kleine, aber doch sehr nützliche Option.

Einen weiteren Nebeneffekt dieses 'Patches' bemerkte ich bei einigen Programmen, die in meinen Auto-Ordnern ihren Dienst versehen. Wenn diese Programme nämlich eine Eingabe (Datum o.ä.) verlangen, so kann man die Daten schon vor dem Start des Programmes (immer hübsch langsam) eingeben. Normalerweise übernimmt das Programm das Eingetippte dann richtig; man sollte die Return-Taste jedoch erst dann betätigen, wenn man die Eingaben kontrolliert hat, denn beim Tippen ohne Bildschirm-Echo kann doch leicht etwas schief gehen...

So erspart man sich unter Umständen etwas Wartezeit. Womit dieser Effekt wohl zusammenhängt? Nun, daß vermag auch ich nicht zu sagen, allerdings schätze ich, daß das Löschen des Bildschirmes auch gleichzeitig den Tastaturpuffer leert. Wie gesagt, reine Spekulation...

Martin Fangmeyer

Byte	Inhalt	Funktion
30	27	Escape (ASCII-Zeichen 27) kennzeichnet einen Grafik-Befehl (Escape-Sequenz)
31	69	Bildschirm löschen, Cursor in Home-Position
32	27	Escape-Sequenz
33	102	Sollte der Text-Cursor auf dem Bildschirm zu sehen sein, so wird er ausgeschaltet
33	27	Escape-Sequenz
35	118	Automatischer 'Überlauf' am rechten Rand aus
36	0	Ende des auszugebenden Strings

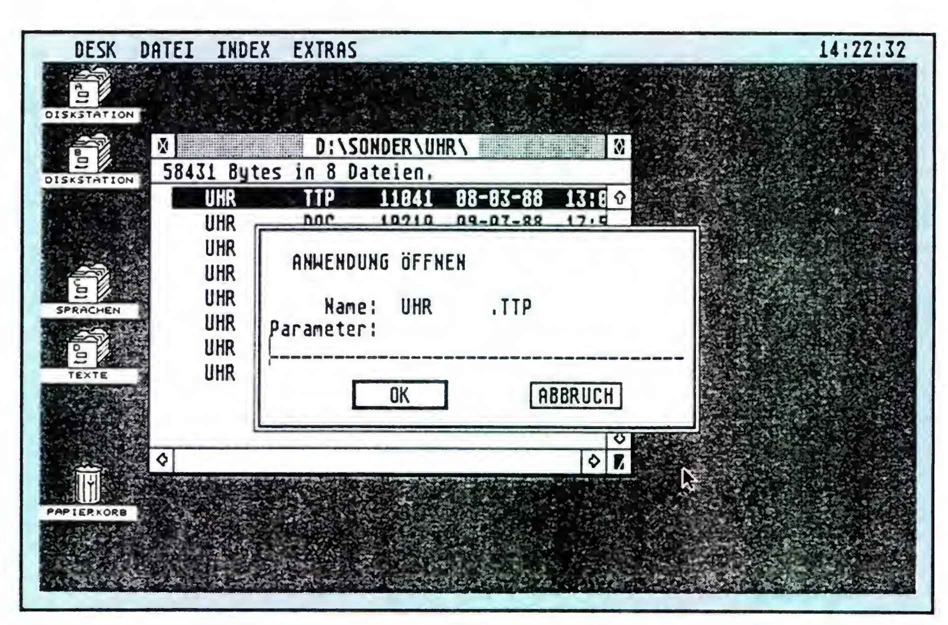
Tabelle : Aufbau des Initialisierungsstrings (ab 30. Byte)

```
1:  Text 60,13,"Bildschirm beim Laden von compiliertem
             GFA-BASIC nicht löschen..."
2:  Do                            ! Endlosschleife
3:    Fileselect "\*.PRG","",A$   ! Datei auswählen
4:    Exit If Not Exist(A$)       ! Schleife verlassen
5:    Open "U",#1,A$              ! Datei zum korrigieren
                                    öffnen
6:    Seek #1,30                  ! Pointer auf 30. Byte
                                    setzen
7:    Out #1,0                    ! Byte löschen
8:    Close                       ! Datei schließen
9:  Loop                          ! Zurück zum Anfang
10: End                           ! Programm beenden
```

"NO_CLS"-Patch für GFA-BASIC

EINE BILDSCHIRM-UHR AM ST

Das hier vorgestellte Listing zeigt Ihnen, wie Sie eine Uhr in der Sprache C selbst programmieren und die selbst überwachte Uhrzeit auf dem Bildschirm anzeigen können, ohne auf vorgefertigte Uhr- und Ausgabefunktionen zurückzugreifen. Neben anderen nützlichen Informationen erfahren Sie außerdem, wie eigene Funktionen in die VBL-Slots eingebunden und wie Systemvariablen benutzt werden.



Man kann sogar die X/Y-Koordinaten für die Bildschirmposition übergeben.

Noch 'ne Uhr, werden einige von Ihnen sagen, denn schon in der Ausgabe 10/86 der ST-Computer-Zeitschrift wurde eine Bildschirmuhr als Listing vorgestellt. Aber warum nicht? Denn wie sich auch Armbanduhren, Wanduhren und Digitaluhren von ihrem Innenleben her unterscheiden, so können sich auch Bildschirmuhren von Ihrer Funktionsweise her unterscheiden.

Daß das hier vorgestellte Listing nicht nur eine interessante Alternative im Hinblick auf die programmtechnische Realisierung einer vom eigenen Programm überwachten Uhrzeit und deren Darstellung auf dem Bildschirm zeigt, sondern auch gleichzeitig einen kleinen Einblick in die Benutzung von Systemvariablen und Interrupts ermöglicht und darüber hinaus auch ohne Schwierigkeiten in eigene Programme eingebunden werden kann, möchte ich Ihnen im folgenden erläutern.

Eine Möglichkeit, die Uhrzeit auf dem Bildschirm darzustellen, wurde schon in der Ausgabe 10/86 vorgestellt, wobei ein Desk-Accessory programmiert wurde, um die quasiparallele Abarbeitung der Uhranzeige zu gewährleisten.

Doch leider wurde der Programmteil zur Berechnung und Darstellung der Systemzeit mit in die evnt_multi()-Schleife eingebaut, was den Nachteil mit sich brachte, daß bei Diskettenzugriffen, die Zeit nicht mehr angezeigt wurde, und somit der ebenfalls programmierte Wecker nicht lief, und manchmal nach Beendigung des Schreib- oder Lesezugriffs die Zeitanzeige auf dem Bildschirm erst dann wieder funktionierte, wenn man das Desk-Accessory erneut mit der Maus aktivierte.

Der Anzeigetakt der dort realisierten Uhr konnte im geringsten Fall zwei Sekunden betragen, die Uhrzeit wurde nur bei Benutzung des Desktops angezeigt. Die Ursachen dieser Probleme liegen bei der Verwaltung der Accessories im Betriebssystem des ST.

Ich möchte mich an dieser Stelle aber nicht über die Accessoryverwaltung auslassen, das könnte vielleicht das Thema eines anderen Artikels werden, sondern nur soviel dazu sagen, daß die Accessories nicht in regelmäßigen Abständen aufgerufen werden und sind daher für die periodische Darstellung einer Uhrzeit nicht unbedingt geeignet.

Wo könnte dann aber eine Uhranzeige eingebunden werden, die die besagten Nachteile nicht mit sich bringt und die die Uhrzeit auch bei Diskettenzugriffen und auch in einem Kommandointerpreter, wie z.B. der DOS-Shell anzeigt?

Einem/einer aufmerksamen Leser/in der Rubriken 'Software' und 'Programmier-Praxis' in der ST-Computer-Zeitschrift wird eine Antwort dazu nicht schwerfallen:

"Natürlich kann ich Funktionen als Interruptroutinen vom Betriebssystem aufrufen lassen, z.B. wenn ich sie in die Liste der VBL-Interrupt-Routinen einfüge!"

Gut, aber das ist, wenn überhaupt, nur die halbe Miete. Denn, wie bekomme ich mein Programm dazu, die Zeit in Sekundentakten anzuzeigen, wenn sogar die XBIOS-Funktion 23 namens 'Gettime()' nur alle zwei Sekunden eine veränderte Zeit ausgibt?

Nachdenken .... irgendwo muß es eine Variable geben, die in einem bestimmten Takt verändert wird, denn wie sollte das Betriebssystem sonst seine periodischen Aufgaben durchführen, wenn es nicht wüßte, wann eine Periode vorüber ist?
Also Suchen bei den Systemvariablen ... such, such, such - aha, da ist sie. _HZ_200 wird sie genannt, steht an der Speicherstelle $4ba und wird 200 Mal in der Sekunde um den Wert 1 erhöht. Also immer wenn _HZ_200 um 200 erhöht wurde, ist eine Sekunde vergangen und eine neue Zeitangabe sollte auf dem Bildschirm erscheinen.
Wir könnten nun zu Anfang des Programms, in der Funktion 'main()', einmal die Zeit mit einer üblichen Funktion wie 'Gettime()', 'getclk()' beim Lattice C-Compiler oder in einer anderen Sprache mit einer anderen Funktion holen, und diese nach jeder vergangenen Sekunde erhöhen. Damit wäre die selbstverwaltete Uhrzeit realisiert.
So ist es auch im Listing geschehen, welches übrigens mit dem Lattice C-Compiler V 3.04 geschrieben wurde. Die Ausgabe unserer internen Programmzeit auf den Bildschirm stellt uns vor ein weiteres, aber auch nicht unlösbares Problem.
Die GEM-Funktion v_gtext() kann nicht benutzt werden, woher soll das 'handle' kommen, welches sie als Parameter benötigt? Ein Versuch, in der Interruptroutine ein neues 'handle' mit der Funktion graf_handle() zu erhalten, endet mit Bombenhagel.
Auch die C-Standard-Bibliotheks-Funktion printf() verhält sich da nicht freundlicher. Abhilfe schafft eine eigene Ausgaberoutine, die uns die Ziffern der Systemzeit direkt in den Bildschirmspeicher schreibt.
Dazu benötigen wir noch die Bytes, aus denen die Ziffern zusammengestellt sind, im Listing ist das der Buffer 'clfont[]', in dem der Reihe nach die Ziffern '0' bis '9' und ein ':' (Doppelpunkt) definiert sind. Eine Ziffer besteht hier aus jeweils 16 Bytes, die, untereinander auf dem Bildschirm ausgegeben, das entsprechende Bild der Ziffer ergeben. So, das Grundprinzip des Programms sollte nun klar sein, begeben wir uns also zum Listing.

Das Listing

Das Listing ist für den Lattice C-Compiler, Version 3.04, geschrieben und sollte sich ohne große Änderungen auch mit anderen C-Compilern compilieren lassen. Beachten Sie, daß die im Listing benutzten Funktionen 'getclk()' und 'stch_i()' der Lattice C-Bibliothek entnommen sind, und bei anderen Compilern wahrscheinlich andere Namen haben.
Zu Anfang werden, wie üblich, die Includedateien und Definitionen für den Präprozessor aufgeführt. Die Bedeutung der darauf folgenden Variablen entnehmen Sie bitte dem Listing.

main()

Hier werden zunächst die an das Programm übergebenen Koordinaten geprüft und die y-Koordinate in Bildschirmzeilen umgerechnet. Dann wird die Adresse des Bildschirms geholt und die tatsächliche Ausgabestelle auf dem Bildschirm berechnet. Mit der Funktion getclk() aus der Lattice C-Bibliothek schreibt man die aktuelle Systemzeit in den Buffer 'clock'. In 'clock[4]' befinden sich die Stunden, in 'clock[5]' die Minuten und in 'clock[6]' die Sekunden.
Danach wird der Inhalt von _HZ_200 der Variablen 'start' zugeordnet, anschließend die Adresse der Funktion 'do_it()' mit der Funktion 'to_vbl_list()' in die Liste der Routinen eingefügt, die bei einem VBL (Vertical Blank) ausgeführt werden sollen.
'to_vbl_list()' holt sich dazu die Anzahl der auszuführenden Funktionen aus der Systemvariablen 'NVBLS' an der Adresse $454 und einen Zeiger auf die Liste dieser Funktionen aus der Systemvariablen '_VBLQUEUE' mit der Adresse $4ce.
Vom Ende der Liste bis zu deren Anfang wird nach einem Wert '0L' gesucht, welches einen freien Slot kennzeichnet, an dem ein Zeiger auf eine eigene Routine eingetragen werden kann. Nach dem Auffinden von '0L' wird die Adresse von 'do_it()' eingetragen und die Funktion mit dem Rückgabewert 'TRUE' beendet. Die Benutzung dieser Funktion gewährleistet, daß schon vorhandene Zeiger nicht überschrieben werden. Am Ende der Funktion 'main()' wird das Programm verlassen, aber im Speicher behalten. Von jetzt an wird nach jedem VBL unsere Funktion 'do_it()' ausgeführt, 'main()' hat keine Aufgaben mehr.
In der Funktion 'do_it()' wird zuerst überprüft, ob mindestens eine Sekunde seit der letzten Überprüfung vergangen ist, indem der Inhalt der Variablen 'start' (ist ja der Inhalt der Systemvariablen _HZ_200 vom letzten vollen Durchlaufen der Funktion 'do_it()') mit dem aktuellen Inhalt von _HZ_200 verglichen wird.
Wenn weniger als eine Sekunde vergangen ist, wird 'do_it()' nicht weiter ausgeführt, ansonsten addiert man die Anzahl der ganzen vergangenen Sekunden mit Hilfe der Funktion 'add_second()' ebenfalls zum Buffer 'clock[4]' und gibt mit der Funktion 'put_to_screen()' die Uhrzeit auf dem Bildschirm aus. Den Vergleichswert für den nächsten Aufruf von 'do_it()' in 'start' korrigiert man mit den vergangenen Sekundenbruchteilen 'diff', da sonst unsere Uhr nach- oder vorgehen würde.
put_to_screen() zerlegt die ersten drei Bytes des Buffers 'clock' in Zehner- und Einerstelle und übergibt diese als Zeiger auf die zugehörige Bytefolge im Buffer 'clfont[]', der hier durch den Zeiger 'dig' repräsentiert wird, an die Funktion 'dig_copy()'. Diese kopiert nun 16 Bytes ab der Adresse, auf die 'dig' zeigt, an die Adresse, auf die 'schirm' zeigt, und addiert zwischen jedem Byte 80 (Anzahl der Zeichen pro Zeile) zu 'schirm'. Dadurch werden die zusammengehörigen Bytes untereinander dargestellt.

Compilieren und Linken

Die Linkdatei und das Makefile für die Lattice C-Umgebung sind mitabgedruckt. Das fertige Programm sollte die Endung '.TTP' erhalten, damit Parameter vom Desktop aus übergeben werden können. Möchten Sie die Routinen in eigene Programme einbinden, sollten Sie der Funktion 'main()' einen anderen Namen geben, z.B. 'uhr()'. Compilieren Sie dann das Programm und linken Sie es mit Ihren eigenen Routinen zusammen.

Starten des Programms

Kopieren Sie das Programm in den Auto-Ordner, wird die Uhranzeige bei jedem Rechnerstart automatisch aktiviert. Wollen Sie das Programm vom Desktop aus starten, sollte es die Endung '.TOS' haben, damit Sie als ersten Parameter die x- Koordinate (zwischen 0 und 71) und als zweiten Parameter die y- Koordinate (zwischen 0 und 24) eingeben können. Übergeben Sie aber keine Koordinaten, so benutzt das Programm die Defaultwerte, die Sie im Quellcode natürlich nach Belieben ändern können.

So, nun bleibt mir noch Ihnen viel Spaß beim Abtippen zu wünschen.

Uwe Gohlke

```
  1:  /****************************************************/
  2:  /*        clock for autostart-ordner, 1/88, v1.0    */
  3:  /*        (c)     uwe gohlke                        */
  4:  /*                burgwedeler str. 26               */
  5:  /*                3004 isernhagen 2                 */
  6:  /*        für st-computer zeitschrift               */
  7:  /****************************************************/
  8:  #include "dos.h"
  9:  #include "osbind.h"
 10:
 11:
 12:  typedef unsigned char UBYTE; /* unsigned 8 bit int */
 13:  typedef short         WORD;  /* signed 16 bit int  */
 14:  typedef long          LONG;  /* signed 32 bit int  */
 15:  typedef void          VOID;
 16:
 17:
 18:  #define TRUE   1
 19:  #define FALSE  0
 20:
 21:
 22:  #define _VBLQUEUE 0x456L
 23:  #define _HZ_200 0x4baL
 24:  #define NVBLS 0x454L
 25:
 26:
 27:  #define FONTSIZE 16
 28:  #define CH_PER_LINE 80
 29:  #define RPS 199
 30:  #define OKARGS 3
 31:  #define NORMXPOS 71
 32:  #define NORMYPOS 0
 33:  #define MAXXPOS 71
 34:  #define MAXYPOS 24
 35:  /****************************************************/
 36:  /*                                                  */
 37:  /****************************************************/
 38:  extern  LONG  _32K;  /* nur für lattice c-compiler */
 39:          LONG  _32K = 0x8000; /* v 3.04 benötigt.   */
 40:
 41:  UBYTE      clock[8],  /* buffer für lokale zeit     */
 42:    *schirm;         /* pointer auf bildschirmspeicher*/
 43:
 44:  LONG       start;     /* zeit der letzten abfrage   */
 45:
 46:  int        posx,posy; /* clockposition              */
 47:
 48:  UBYTE      clfont[]={ /* fontdaten                  */
 49:    0,0,0x3c,0x7e,0x66,0x66,0x66,0x6e,0x76,0x66,0x66,
         0x66,0x7e,0x3c,0,0,
 50:    0,0,0x18,0x18,0x38,0x38,0x18,0x18,0x18,0x18,0x18,
         0x18,0x7e,0x7e,0,0,
 51:    0,0,0x3c,0x7e,0x66,0x66,0x0c,0x0c,0x18,0x18,0x30,
         0x30,0x7e,0x7e,0,0,
 52:    0,0,0x7e,0x7e,0x0c,0x0c,0x18,0x18,0x0c,0x0c,0x66,
         0x66,0x7e,0x3c,0,0,
 53:    0,0,0x0c,0x0c,0x1c,0x1c,0x3c,0x3c,0x6c,0x6c,0x7e,
         0x7e,0x0c,0x0c,0,0,
 54:    0,0,0x7e,0x7e,0x60,0x60,0x7c,0x7e,0x06,0x06,0x06,
         0x66,0x7e,0x3c,0,0,
 55:    0,0,0x1c,0x3c,0x70,0x60,0x60,0x7c,0x7e,0x66,0x66,
         0x66,0x7e,0x3c,0,0,
 56:    0,0,0x7e,0x7e,0x06,0x06,0x0c,0x0c,0x18,0x18,0x30,
         0x30,0x30,0x30,0,0,
 57:    0,0,0x3c,0x7e,0x66,0x66,0x3c,0x3c,0x66,0x66,0x66,
         0x66,0x7e,0x3c,0,0,
 58:    0,0,0x3c,0x7e,0x66,0x66,0x7e,0x3e,0x06,0x06,0x06,
         0x0e,0x3c,0x38,0,0,
 59:    0,0,0x00,0x00,0x18,0x18,0x18,0x18,0x00,0x00,0x18,
         0x18,0x18,0x18,0,0
 60:  };
 61:
 62:
 63:  VOID       do_it(),put_to_screen(),dig_copy(),
             add_second();
 64:  WORD       to_vbl_list();
 65:  /****************************************************/
 66:  /* installiere die funktion 'do_it' in die vbl-     */
 67:  /* funktionenliste, setze koordinaten für uhren-    */
 68:  /* ausgabe, hole die systemzeit & ordne sie dem     */
 69:  /* buffer 'clock' zu, verlasse die funktion (das    */
 70:  /* programm), aber behalte es im speicher           */
 71:  /****************************************************/
 72:  main(argc,argv)
 73:  UBYTE   **argv;
 74:  int     argc;
 75:  {
 76:    LONG    oldstack;
 77:
 78:
 79:    if(argc==OKARGS) {         /* koordinaten setzen  */
 80:       stch_i(argv[1],&posx); /* wandle string ->int */
 81:       stch_i(argv[2],&posy);
 82:       if(posx<0)             /* teste koordinaten */
 83:             posx*=-1;        /* auf gültigkeit    */
 84:       if(posy<0)
 85:             posy*=-1;
 86:       if(posx>MAXXPOS)
 87:             posx=MAXXPOS; /* defaultwert         */
 88:          if(posy>MAXYPOS)
 89:                posy=MAXYPOS;
 90:    }
 91:    else {                       /* defaultposition   */
 92:          posx=NORMXPOS;
 93:          posy=NORMYPOS;
 94:    }
 95:    posy*=CH_PER_LINE*FONTSIZE; /* konvertiere zu
                                     linienhöhe*/
 96:    printf("Uhranzeige aktiv! (c) uwe gohlke 1/88\n");
 97:    schirm=Logbase()+posx+posy; /* bildschirmadresse */
 98:    getclk(clock);            /* hole systemzeit     */
 99:    oldstack=Super(0L);       /* supervisormodus an  */
100:    start=*(LONG *) _HZ_200;  /* 200 hz zaehler      */
101:    Super(oldstack);          /* supervisormodus aus */
102:    to_vbl_list(&do_it);      /* in vbl-liste einordnen*/
103:    Ptermres(15000L,0);       /* programm verlassen & */
104:                              /* im speicher behalten */
```

```
105: }
106: /**************************************************/
107: /*                                                */
108: /**************************************************/
109: VOID
110: do_it()
111: {
112:   register        diff,
113:                   noch,
114:                   neu;
115:
116:
117:   if((diff=((neu=*(LONG *)_HZ_200))-start)>=RPS)
       {                        /* prüfe, ob            */
118:                            /* mindestens 1 sekunde */
119:                            /* seit dem letzten     */
120:                            /* aufruf vergangen ist.*/
121:                            /* aufruf.              */
122:          noch=diff/RPS;    /* anzahl der vergan-   */
123:                            /* genen sekunden.      */
124:          diff-=noch*RPS;   /* sekundenbruchteil.   */
125:          start=neu-diff+1; /* start für naechsten */
126:          while(noch--)     /* addiere sekunden zur */
127:                  add_second(&clock[4]); /* lokalen
                                           zeit.  */
128:          put_to_screen(schirm,&clock[4],clfont);
                               /* ausgabe auf schirm   */
129:   }
130: }
131: /**************************************************/
132: /*  addiere 1 sekunde zum zeitbuffer 'clock' und  */
133: /*  aktualisiere gegebenenfalls die minuten und
        stunden                                       */
134: /**************************************************/
135: VOID
136: add_second(clock)
137: UBYTE    *clock;
138: {
139:   if(++*(clock+2)==60) {          /* sekunden */
140:          *(clock+2)=0;
141:          if(++*(clock+1)==60) {    /* minuten */
142:                  *(clock+1)=0;
143:                  if(++*clock==24) { /* stunden */
144:                          *clock=0;
145:                  }
146:          }
147:   }
148: }
149: /**************************************************/
150: /*      schreibe 'clock' zur bildschirmposition   */
151: /*      'schirm' mit den zeichen*/ in 'dig'.      */
152: /**************************************************/
153: VOID
154: put_to_screen(schirm,clock,dig)
155: UBYTE     *clock,
156:   *schirm,
157:   *dig;
158: {
159:   register        i,
160:                   ten,
161:                   one;
162:
163:   for(i=0; i<3; i++,clock++) { /* 3: minuten,     */
164:                           /* sekunden, stunden */
165:          ten=(*clock)/10;     /* dezimalstelle     */
166:          dig_copy(schirm,dig+ten*FONTSIZE);
                                  /* auf schirm        */
167:          schirm++;
168:          one=(*clock)-ten*10;
169:          dig_copy(schirm,dig+one*FONTSIZE);
170:          schirm++;
171:          if(i<2) {     /* schreibe ':' dazwischen*/
172:                  dig_copy(schirm,dig+10*FONTSIZE);
173:                  schirm++;
174:          }
175:   }
176: }
177: /**************************************************/
178: /*  kopiere FONTSIZE bytes vom buffer 'dig' zur
        adresse 'schirm'                              */
179: /*  addiere zwischen jedem byte CH_PER_LINE zu
        'schirm'                                      */
180: /**************************************************/
181: VOID
182: dig_copy(schirm,dig)
183: UBYTE      *schirm,
184:   *dig;
185: {
186:   register        i;
187:
188:   for(i=0; i<FONTSIZE; i++,dig++) {
189:          *(schirm+i*CH_PER_LINE)=*dig;
190:   }
191: }
```

```
192: /**************************************************/
193: /* schreibe den wert 'rein' in die vbl-routinen-
        liste an eine stelle, an der eine '0L' steht   */
194: /* und gib TRUE(1) zurück                         */
195: /* wenn keine '0L' vorhanden, gib FALSE (0) zurück */
196: /**************************************************/
197: WORD
198: to_vbl_list(rein)
199: register  rein;
200: {
201:   register        i;
202:   register        oldstack;
203:   LONG            *inh;
204:
205:
206:   oldstack=Super(0L);       /* supervisor-mode an  */
207:   inh=*(LONG *) _VBLQUEUE; /* zeiger auf vblqueue */
208:   for(i=0,inh+=(*(WORD *) NVBLS); i< *(WORD *) NVBLS;
         i++,inh--) {
209:          if(!*inh) {        /* wenn inhalt 0        */
210:                  *inh=rein; /* neuer inhalt rein */
211:                  Super(oldstack);
                             /* supervisormode aus    */
212:                  return(TRUE);
213:          }
214:   }
215:   Super(oldstack);          /* usermode aus         */
216:   return(FALSE);
217: }
```

ENDE

BASIC-SHELL

Übersichtlicher Zugriff auf Interpreter und Compiler

Über diese BASIC-Shell kann man auf einfache Art und Weise den GFA-Interpreter und -Compiler benutzen. Man erspart sich das lästige Suchen im Inhaltsverzeichnis und kann so auf Tastendruck das gewünschte Programm laden und starten. Natürlich funktioniert das auch mit anderen Programmen wie in unserem Beispiel das OMIKRON-BASIC.

Das Programm besteht im wesentlichen aus vier Teilen. Der erste Teil (Zeilennummer 1-7) trifft die wichtigsten Vorbereitungen für das gesamte Programm. VOID FRE(0) erlaubt es dem Compiler, einen besonders effektiven Kode zu produzieren - wie auch im GFA-BASIC-Handbuch auf Seite F 34 beschrieben. Dieser Befehl hat für das Programm ansonsten keine Bedeutung. Wichtiger ist hingegen die Zeile 2, die 10000 Bytes für das Programm reserviert und dadurch dafür sorgt, daß es nicht selber durch die gestarteten Programme zerstört wird.

Die Tex$()-Variable hat eine zentrale Rolle im Programm. Sie enthält neben dem Text für den Monitor (GFA_BASIC { Editor...}) standardmäßig eine "0" für ein existierendes Programm (Näheres im 3. Teil) und den Dateinamen inklusive des Pfades, da von verschiedenen Laufwerken (A und O) geladen wird. Diese eine Variable enthält also drei Variablen. Das habe ich so gelöst, damit nicht unnötig viele Variablenfelder aufgebaut werden müssen.

Der zweite Teil (Zeilennummern 8-24 und 37-39) sorgt für den Bildschirmaufbau. also den großen Kasten in der Mitte mit dem Programmnamen "BASIC_SHELL" und dem fünften Menüpunkt, der Beendigung des Programmes, denn dieser ist immer anwählbar.

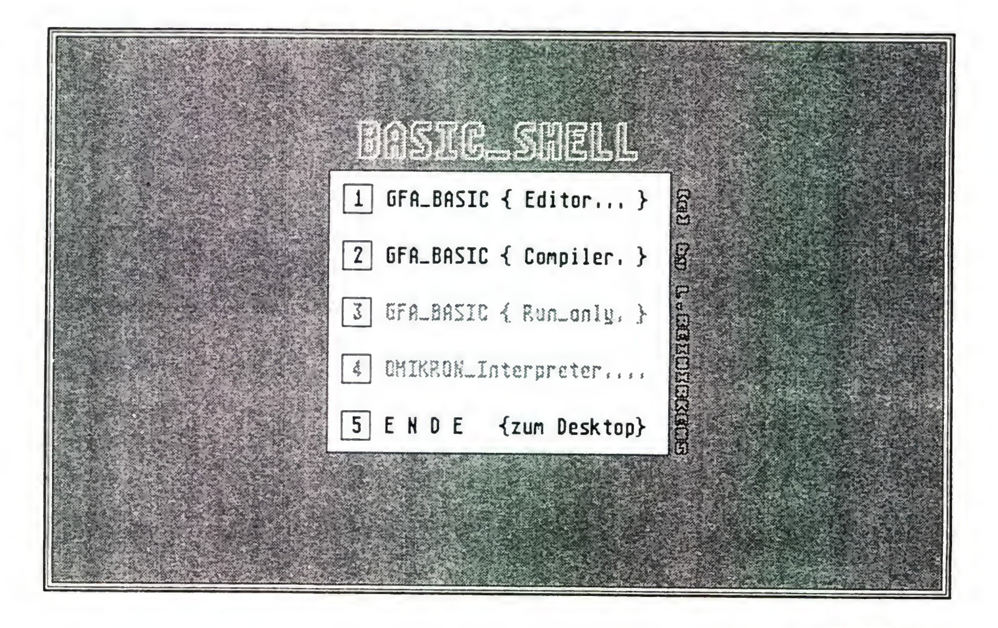
Der dritte Teil ist mit im zweiten eingebaut (Zeilennummer 25-36). Hier werden die Programmnamen aufgelistet und gleichzeitig kontrolliert, ob das entsprechende Programm überhaupt anwählbar ist, d.h. ob sich die richtige Diskette im Laufwerk befindet. Dieses wird durch den Befehl EXIST("Dateiname") (s. GFA-Handbuch Anhang A) ermöglicht. Ist das Programm nicht auf der Diskette, ist also "IF NOT EXIST(...) THEN" wahr, wird die Textausgabe auf hell (Zeile 27) und die Teilvariable von Tex$() von "0" auf "1" geändert (Zeile 28). Ist das Programm vorhanden, wird die Teilvariable von Tex$() auf "0" gesetzt. Das 24. Zeichen der Variablen Tex$() gibt also Auskunft darüber, ob die Datei geladen und gestartet werden kann.

Im vierten und letzten Teil kann man das Programm auswählen, mit dem man gerne arbeiten möchte. Dazu muß man die Variable I$ löschen (Zeile 41) und warten, bis eine Taste gedrückt wird (Zeile 42-44). Der Inhalt der Taste ist I$ bzw. der Wert (Zahl) in I% (Zeile 45).

Ich habe mich absichtlich auf eine einfache Tastatureingabe beschränkt, da das Programm selber nicht zu kompliziert werden sollte, und dieses für diesen Zweck voll und ganz genügt. Darum habe ich auch die Maus in Zeile 9 ausgeschaltet, damit der Pfeil nicht irritiert.

Falls eine "5" gedrückt wird, erfolgt der Abbruch des Programms. (Zeile 46-48). Wenn eine ungültige Taste gedrückt wird, also ein Buchstabe oder Sonderzeichen, eine "0" oder eine Zahl größer als fünf, wird wieder auf eine neue Tastatureingabe gewartet.

Ist jetzt eine Zahl zwischen "1" und "4" gedrückt worden, kommt wieder das 24. Zeichen zur Geltung. Ist es gleich "1", also Datei nicht vorhanden (Zeile 52), wird kein Programm gestartet, sondern es wird auf eine neue Tastatureingabe gewartet (GOTO Starten_2). Mit IF BIOS(9,0)>0 THEN wird gleichzeitig noch überprüft, ob nicht zwischenzeitlich die Diskette gewechselt wur-



de. Wenn ja, dann muß der Bildschirm neu aufgebaut und das 24. Zeichen in Tex$() korrigiert werden, d.h. prüfen, ob die Dateien noch immer auf einer Diskette im Laufwerk sind (Zeile 54 GOTO starten_1).
Ist jetzt ein Programmstart möglich, wird dieser in Zeile 60 durch den Befehl EXEC 0,"Dateiname","","" erledigt.
EXEC flg,nam,cmd,env: "flg" kann "0" oder "3" sein. In diesem Programm ist jedoch nur "0" sinnvoll, da das Programm nicht nur geladen (flg=3), sondern auch gestartet werden soll. "nam" ist der Dateiname. "cmd" ist eine zu übergebende Kommandozeile, auf die im Normalfall verzichtet werden kann; daher ein Leerstring "". "env" ist der Environment-String, der jedoch auch hier unwichtig ist.

Lars Reinirkens

Listing für die Basic-Shell

```
  1:  Void Fre(0)
  2:  Reserve 10000
  3:  Dim Tex$(3)
  4:  Tex$(0)="GFA_BASIC { Editor... }0A:\GFABASIC.PRG"
  5:  Tex$(1)="GFA_BASIC { Compiler. }0A:\GFA_BCOM.PRG"
  6:  Tex$(2)="GFA_BASIC { Run_only. }0A:\GFABASRO.PRG"
  7:  Tex$(3)="OMIKRON_Interpreter....00:\OM-BASIC.PRG"
  8:  Starten_1:
  9:  Hidem
 10:  Cls
 11:  Deffill 1,2,4
 12:  Box 1,1,638,398
 13:  Box 2,2,637,397
 14:  Box 4,4,635,395
 15:  Pbox 7,7,632,392
 16:  Deffill 1,0,0
 17:  Pbox 200,100,439,299
 18:  Graphmode 2
 19:  Deftext 1,17,2700,6
 20:  Text 445,110,"(c) by L.REINIRKENS"
 21:  Deftext 0,16,0,32
 22:  Text 220,90,"BASIC_SHELL"
 23:  Graphmode 0
 24:  Deftext 1,0,0,13
 25:  For T%=0 To 3
 26:    If Not Exist(Right$(Tex$(T%),15)) Then
 27:      Deftext 1,2,0,13
 28:
Tex$(T%)=Left$(Tex$(T%),23)+"1"+Right$(Tex$(T%),15)
 29:    Else
 30:
Tex$(T%)=Left$(Tex$(T%),23)+"0"+Right$(Tex$(T%),15)
 31:    Endif
 32:    Box 210,110+T%*40,230,130+T%*40
 33:    Text 217,125+T%*40,T%+1
 34:    Text 240,125+T%*40,Left$(Tex$(T%),23)
 35:    Deftext 1,0,0,13
 36:  Next T%
 37:  Box 210,270,230,290
 38:  Text 217,285,5
 39:  Text 240,285,"E N D E   {zum Desktop}"
 40:  Starten_2:
 41:  I$=""
 42:  While I$=""
 43:    I$=Inkey$
 44:  Wend
 45:  I%=Val(I$)
 46:  If I%=5
 47:    End
 48:  Endif
 49:  If I%<1 Or I%>4 Then
 50:    Goto Starten_2
 51:  Else
 52:    If Val(Mid$(Tex$(I%-1),24,1))=1 Then
 53:      If Bios(9,0)>0 Then
 54:        Goto Starten_1
 55:      Else
 56:        Goto Starten_2
 57:      Endif
 58:    Else
 59:      Showm
 60:      Exec 0,Right$(Tex$(I%-1),15),"",""
 61:    Endif
 62:  Endif
 63:  Goto Starten_1
```

ENDE

Betr.: Programmier Praxis

Die "Progammier Praxis" ist ein fester Bestandteil der ST Computer geworden. Gedacht ist sie für all diejenigen, die ihren Rechner selbst programmieren, und dazu Tips und Anregungen gebrauchen können. Diese Rubrik kann aber nur dann bestehen, solange viele Leser (Sie eingeschlossen) sich daran beteilligen. Wir fordern Sie deshalb auf, Ihre Ideen in einen Umschlag zu stecken und auf dem schnellsten Weg zu uns zu schicken. Voraussetzungen gibt es praktisch keine: es muß lediglich interessant sein und so dokumentiert, daß auch andere Anwender daraus einen Nutzen ziehen können. Auch bei den Sprachen gibt es keinerlei Einschränkungen: PASCAL, C, BASIC; Assembler, Modula 2, APL, Fortan, LISP, Prolog, PEARL,... und sogar LOGO ist erwünscht.
Veröffentliche Programme werden natürlich angemessen honoriert. Bitte beachten Sie bei der Einsendung: Schicken Sie den Quelltext und das evtl. compilierte Programm, sowie die Dokumentation auf Papier und Diskette. Die Diskette wird zurückerstattet.
Einsendungen direkt an die Redaktion:

"MERLIN" Computer GmbH
ST Computer Redaktion
´Programmier Praxis´
Industriestraße 26
D-6236 Eschborn Tel.: 0 61 96 / 48 18 11



ABO

ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Einzelheft- u. Monatsdisketten Bestellung



Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Kleinanzeigen



Kleinanzeigen

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Abonnement



Ja, bitte senden Sie mir die ATARI-Computer Fachzeitschrift ab ______________
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,– frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,– Normalpost, DM 120,– Luftpost)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name

Vorname

Straße/Nr.

PLZ Ort

Datum Unterschrift

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. BLZ

Institut Ort

☐ Ein Verrechnungsscheck über DM ______ liegt bei.

☐ Vorauskasse per Post-Einzahlung (Zahlkarte)

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum Unterschrift



ABO



Einzelheft- u. Monatsdisketten Bestellung



ST-Computer können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 6,— (1986 + 1987) DM 7,— (1988) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli/Aug.	Sept.	Okt.	Nov.	Dez.	
											1986 = DM
											1987 = DM
											1988 = DM

+ Gebühr für Porto u. Verp. = DM

(1 Heft DM 2,–, ab 2 Hefte DM 5,–)

☐ Scheck in Höhe zus. DM liegt bei

Disketten Service

Alle Programme, die in ST-Computer veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 ST-Computer-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

Preis je Diskette 28,– DM	Jan./Febr.	März/Apr.	Mai/Juni	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.	Mai/Juni
	87	87	87	87	87	87	88	88	88

Lieferung: gegen beigefügten Scheck zuzügl. 5,— DM Versandkosten, unabhängig der bestellten Menge.



Einzelheft- u. Monatsdisketten Bestellung



Kleinanzeigen-Auftrag

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an ☐ Hardware ☐ Software

Ich suche ☐ Hardware ☐ Software

☐ **Tausch** ☐ **Kontakte**

☐ Verschiedenes

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

☐ privat = DM 7,– je Zeile incl. MwSt.
☐ gewerblich = DM 15,– je Zeile + MwSt.
☐ Chiffregebühr = DM 10,–

☐ Scheck über DM ______ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum Unterschrift

Absenderangaben auf der Rückseite nicht vergessen



Kleinanzeigen



Kontaktkarte

Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

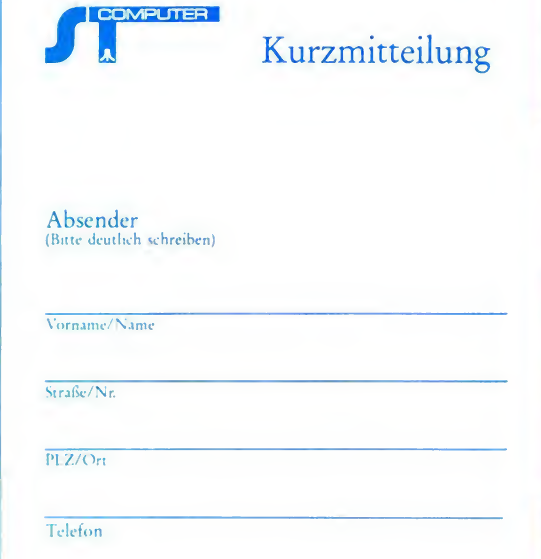
Bitte freimachen

Firma

Straße/Postfach

PLZ Ort

Kurzmitteilung



Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

Postkarte

Bitte freimachen

Merlin Computer GmbH
ST-Computer Redaktion
Industriestraße 26

6236 Eschborn



PD Bestellung



PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

Merlin Computer GmbH
ST-Computer Redaktion
Industriestraße 26

6236 Eschborn

ST COMPUTER

☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in ST-Computer Heft ______ Seite ____

Menge	Produkt/Bestellnummer	DM	gesamt DM

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Abgesandt am:

ST COMPUTER

Firma:

Bemerkungen:

ST COMPUTER

Meine Meinung

Zu dem Artikel ____________ in Heft ____, Seite ____
hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten, GEM/TOS)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am ST, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der ST-Computer werden. Meine Fachgebiete: (z.B. LISP, Pearl, Modula-2, Assembler ...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11

ST COMPUTER

Kurzmitteilung

ST COMPUTER

PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,– bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum

Unterschrift

ST COMPUTER

PD Bestellung

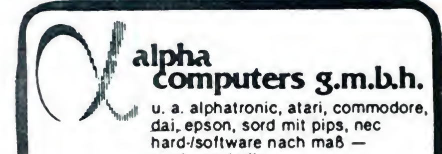
Datum

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

Anzeigenschluß für Heft 7-8/88 ist der 13.05.88

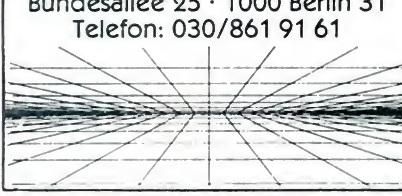
1000 Berlin



u. a. alphatronic, atari, commodore, dai, epson, sord mit pips, nec
hard-/software nach maß — servicetechnik
Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/891 1082

Computare
Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d

DATAPLAY
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61





. . . wir machen Spitzentechnologie preiswert.



DIGITAL COMPUTER
Verkaufsbüro (1.OG) 1 Berlin 12
Knesebeckstr. 76 · Tel. 882 77 91
Software Hardware Beratung Zubehör Service Literatur

Ihre Tür zur Zukunft:
KARSTADT
computer—center
hardware·software·problemlösungen
Berlin, Hermannplatz, Telefon (0 30) 6 90 81

1000 Berlin





Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 26 111 26

Computershop Edith Behrendt
Fürbringerstraße 26 · 1000 Berlin 61
Telefon 0 30/691 76 66



. . . wir machen Spitzentechnologie preiswert.
Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61



2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 44 00

Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 0 40/641 50 91

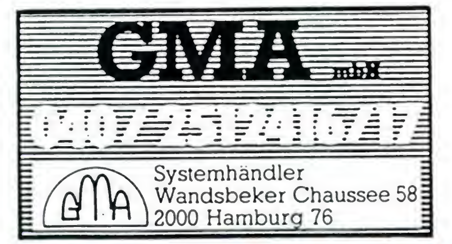
Hardware
Software
Beratung
Service



ATARI Systemfachhändler
Münsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1

Gerhard u. Bernd Waller GbR
Computer & Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
☎ 040/570 60 07 + 570 52 75

NEU: Software Shop
RADIX Bürotechnik
Heinrich Barth Str. 13
2000 Hamburg 13
Telefon: 0 40-44 16 95



2000 Norderstedt



2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt
Heiligengeiststr. 20, 2120 Lüneburg
Tel. 04131/46122, Btx 40 24 22
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Bübel Martin Kopplow

Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 04 31/56 70 42

2350 Neumünster



2390 Flensburg

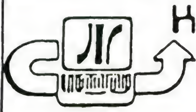


2800 Bremen

PS-DATA
Doventorsteinweg 41
2800 Bremen
Telefon 04 21 - 17 05 77

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann
„Bürger" 160
2850 Bremerhaven
Tel. 04 71/4 20 06

HARDWARE · SOFTWARE · PAPIERWARE

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler

Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

3000 Hannover

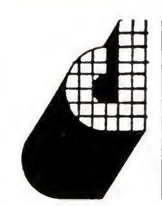
COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36

• Software • Hardware • Organisation • Beratung • Schulung

Computer GmbH PCH

Großer Hillen 6 · 3000 Hannover 71
Telefon (0511) 52 25 79



DATALOGIC COMPUTERSYSTEME
ATARI ST-COMPUTER HARDWARE SOFTWARE
BERATUNG SERVICE VERKAUF
CALENBERGER STR. 26
3000 HANNOVER 1
TEL 0511 - 32 64 89

trendDATA Computer
IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc.

trendDATA Computer GmbH
Am Marstall 18–22 · 3000 Hannover 1
Telefon (05 11) 1 66 05-0

3040 Soltau

F & T Computervertrieb

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91/1 65 22

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software

Woltorfer Str. 8, 3150 Peine
Tel. 0 51 71/60 52/3 o. 0 51 73/79 09

3170 Gifhorn

COMPUTER-HAUS GIFHORN
Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

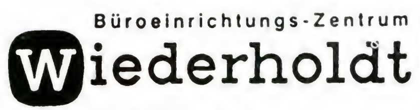
3300 Braunschweig

COMPUTER STUDIO BRAUNSCHWEIG

Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77/78

3400 Göttingen

Büroeinrichtungs-Zentrum



3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51/38 57-0

3470 Höxter



3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Tel. (05 61) 70 00 00

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

4000 Düsseldorf

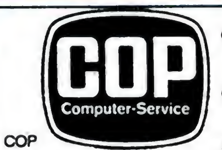
HOCO
EDV ANLAGEN GMBH
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4050 Mönchengladbach



Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 187 64

4130 Moers



- Service-Center
- Hardware
- Software
- Erweiterungen

COP
Computer Service GmbH
Essenberger Str. 2H
4130 Moers
Tel. 02841 / 23 58 5

4150 Krefeld



- Service-Center
- Hardware
- Software
- Erweiterungen

COP
Computer Service GmbH
Lewerentz Str. 111
4150 Krefeld
Tel. 0 21 51/77 30 42

4250 Bottrop

Megateam-Computer-Systeme
Kirchhellenerstraße 262
4250 Bottrop

4300 Essen

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4320 Hattingen



4330 Mülheim



Computer und Bürotechnik
Vertriebsgesellschaft mbH
Dickswall 79 4330 Mülheim Telefon 02 08/3 40 34

Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene

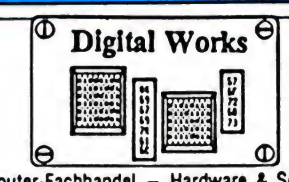
Tandon Computer · VICTOR · SEL-Fernkopierer · NEC · ATARI · OKI

4400 Münster



COMPUTER SYSTEME GMBH
Daimlerweg 39 - 4400 Münster
Telefon 02 51 / 71 99 75 - 9

4410 Warendorf



Computer-Fachhandel – Hardware & Software
Jörg Kellert – Helmut Müller GbR
Brünebrede 17 · 4410 Warendorf
Tel. 0 25 81/6 11 26

4422 Ahaus

ATARI · Epson · Fujitsu
Molecular · NCR · Tandon · Schneider · Star
OCB

OCB-Computershop	OCB-Hard- und Software
Wallstraße 3	Wessumerstraße 49
4422 Ahaus	4422 Ahaus
Tel. 0 25 61/50 21	Tel. 0 25 61/50 21

4430 Steinfurt

ATARI SCHNEIDER STAR NEC SEIKOSHA PANASONIC EPSON

Computer
Büromaschinen
Service
Telefon 02551/2555

Tecklenburger Str. 27 · 4430 Steinfurt

4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120·4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

4600 Dortmund

Bürostudio BOLZ
Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31 - 52 77 13-16

4600 Dortmund





Atari, Genie, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software·Hardware·Beratung
Service·Eilversand

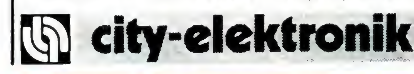
Ihre Ansprechpartner:
v. Schabilnski
Jan P. Schneider

Elisabethstraße 5
4600 Dortmund 1
T. 0231/528184 · Tx 822631 cccsd

Elektronik
Computer
Fachliteratur

ATARI-System-Fachhändler

4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84



4620 Castrop-Rauxel

R. Schuster Electronic

ATARI
System-Fachhändler

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4700 Hamm

computer center



4800 Bielefeld

hardware
software
organisation
service
CSF

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

6300 Gießen



Schneider ATARI Commodore

BAUMS

BÜRO · ORGANISATION

Bahnhofstr. 26 · 6300 Gießen

Telefon: 06 41 / 7 10 96



Thomas Heß

Asterweg 10 · 6300 Gießen

Telefon 06 41 / 3 91 53

6330 Wetzlar



Fachmarkt für Computer u. Unterhaltungs-electronic in Wetzlar,

Einkaufszentrum Bahnhofstraße, Tel. (0 64 41) 4 85 66

6400 Fulda

Schneider ATARI Commodore

WEINRICH

BÜRO · ORGANISATION

Ronsbachstraße 32 · 6400 Fulda

Telefon: 06 61 / 4 92-0

6457 Maintal

Landolt-Computer

Beratung · Service · Verkauf · Leasing

Wingertstr. 112

6457 Maintal/Dörnigheim

Telefon 0 61 81 - 4 52 93

6500 Mainz

:ELPHOTEC

Computer Systeme

Ihr Atari Systemhändler mit eigenem Service-Center

Walpodenstraße 10

6500 Mainz

Telefon 0 61 31 - 23 19 47

6520 Worms

ORION

Computersysteme GmbH

Friedrichstraße 22

6520 WORMS

Tel. 0 62 41 / 67 57 - 67 58

6700 Ludwigshafen

MKV Computermarkt

Bismarck-Zentrum

6700 Ludwigshafen

Telefon 06 21 - 52 55 96

6720 Speyer

MKV Computermarkt

Gilgenstraße 4

6720 Speyer

Telefon 0 62 32 - 7 72 16

6750 Kaiserslautern

C.O.S

Computer Organisation GmbH

Friedrichstr. 4 · 6750 Kaiserslautern

Telefon (06 31) 9 20 51

6800 Mannheim

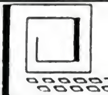


Computersysteme + Textsysteme

6800 Mannheim 24

Casterfeldstraße 74-76

☎ (06 21) 85 00 40 · Teletex 6 211 912



Computer-Center

am Hauptbahnhof GmbH

L 14, 16-17

6800 Mannheim 1

Tel. (06 21) 2 09 83 / 84

6900 Heidelberg

JACOM COMPUTERWELT

Hardware · Software

Schulung · Service

Mönchhofstraße 3 · 6900 Heidelberg

Telefon 0 62 21 / 41 05 14 - 550

HEIDELBERGER COMPUTER CENTER

Bahnhofstraße 1

6900 Heidelberg

Telefon 06221/27132

7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler

ATARI ST



Matrai Computer GmbH

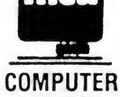
Bernhauser Str 8

7022 L-Echterdingen

☎ (07 11) 79 70 49

7030 Böblingen

Verkauf – Service – Software



Norbert Hlawinka

Sindelfinger Allee 1

7030 Böblingen

Tel. 0 70 31 / 22 60 15

COMPUTER CENTER

7100 Heilbronn

Unser Wissen ist Ihr Vorteil

Walliser & Co.

Mönchseestraße 99

7100 Heilbronn

Telefon 07131/60048



Computer-Welt

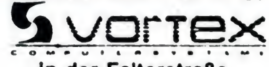
Am Wollhaus 6

7100 Heilbronn

Tel. 0 71 31 - 6 84 01 - 02

7101 Flein

der **COMPUTERLADEN** von



In der Falterstraße

7101 Flein

Beratung, Service, Verkauf, Software-Entwicklung direkt beim Peripherie-Hersteller

7150 Backnang



7400 Tübingen

Werner Brock

COMPUTERSSTUDIO

Poststraße 2 – 4 · D-7400 Tübingen

Tel. 0 70 71 - 3 43 48 · Fax 0 70 21 - 3 47 92

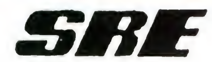
Autorisierter Systemfachhändler für:

ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

7410 Reutlingen

Werner Brock
COMPUTERSHOP
Federnseestr. 17 · 7410 Reutlingen
Tel. 0 71 21 - 3 42 87
Tx 1 72 414 024 RMI D · box:rmi:taisoft · Fax 0 71 21 - 33 97 79
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

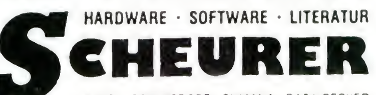
7450 Hechingen



Gesellschaft für Datenverarbeitung mbH
Computer · Drucker
Zubehör · Fachliteratur
Schloßplatz 3 · 7450 Hechingen
Telefon 0 74 71 / 1 45 07

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR



ATARI COMMODORE CUMANA DATA BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON
7475 Meßstetten 1 · Hauptstraße 10 · 0 74 31 / 6 12 80

7480 Sigmaringen

Wir stellen uns für Sie auf den Kopf!
Wir führen:
Hardware - Software - Zubehör
Zeitschriften - Bücher
Wir sind autorisierte Fachhändler von:
ATARI Schneider
Tandon
SOFT & EASY
COMPUTER GMBH
7480 Sigmaringen · Am Rappgassle · Tel 0 75 71 / 1 24 83
7968 Saulgau · Pfarrstraße 13 · Tel 0 75 81 25 98

7500 Karlsruhe

papierhaus erhardt
Am Ludwigsplatz · 7500 Karlsruhe
Tel. 0721-1608-16

MKV GMBH
Kriegsstraße 77
7500 Karlsruhe
Telefon 07 21 - 8 46 13

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk
In der Jeuch 3
7600 Offenburg
Telefon 07 81 / 5 79 74

7640 Kehl/Rhein



Badstrasse 12
Tel. 0 78 07 / 8 22
Telex: 752 913
7607 NEURIED 2
Filiale:
Hauptstrasse 44
Tel. 0 78 51 / 18 22
7640 KEHL/RHEIN
ELEKTRO-MÜNTZER GmbH

7700 Singen

U. MEIER
Computersysteme
7700 Singen-Htwl.
Am Posthalterswäldle 8
Telefon 0 77 31 - 4 42 11

ST-Einkaufsführer
aktuell
preiswert
werbewirksam
Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 0 77 20 / 3 80 71-72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER
computer · fachgeschäft
rösler
Rheingutstr. 1 · 0 75 31 - 2 18 32

7800 Freiburg

CDS
EDV-Service GmbH
Windausstraße 2
7800 Freiburg
Tel. 07 61 - 8 10 47

7800 Freiburg

PYRAMID COMPUTER GMBH
KARTÄUSERSTRASSE 59
D-7800 FREIBURG/BRST.
TELEFON 07 61 - 38 20 38

7890 Waldshut-Tiengen

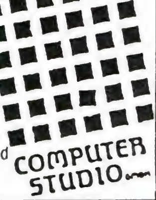
hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 07 31 / 6 26 99

EDV-Systeme
Software-erstellung
Schulung

Systemhaus:
Frauenstr. 28
7900 Ulm/Donau
Tel. 07 31 / 2 80 76
Telex 712 973 csulm-d



7918 Illertissen

bidtech gmbh
technische Informationssysteme
Computerladen
Marktplatz 13
7918 Illertissen
0 73 03 / 50 45

7950 Biberach

HARD AND SOFT COMPUTER GMBH
Biberachs großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Schulstraße 6 · Bei der VHF
7950 Biberach · Tel. 0 73 51 / 1 22 21

7980 Ravensburg

GRAHLE
Expert Grahle Computer
Eisenbahnstr. 33
7980 Ravensburg
Tel.: 0751 / 15955
Vertragshändler für ATARI, Schneider und Star

8000 München



COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

schulz computer
Schillerstraße 22
8000 München 2
Telefon (0 89) 59 73 39
Beratung · Verkauf · Kundendienst

8032 Gräfelfing



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464,851043

8100 Garmisch-Part.

Uwe Langheinrich Elektronik Center
Hindenburgerstr. 45
8100 Garmisch-Partenkirchen
Tel. 0 88 21-7 15 55
Bitte Gratisliste anfordern

Werben Sie im ST COMPUTER EINKAUFSFÜHRER
Rufen Sie uns an.
Heim Verlag 0 61 51/56057

8150 Holzkirchen

ATARI
Besuchen Sie uns!
Fordern Sie unseren Software-Katalog (520ST) an!



MÜNZENLOHER GMBH
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (0 80 24) 18 14

8170 Bad Tölz

Uwe Langheinrich Elektronik Center
Wachterstr. 3
8170 Bad Tölz
Tel. 0 80 41-4 15 65
Bitte Gratisliste anfordern

8330 Eggenfelden

Hot Space
Computer-Centrum
R. Lanfermann
Schellenbruckstraße 6
8330 Eggenfelden
Telefon 0 87 21/65 73
Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71/7 16 10
Innstraße 4
8341 Simbach
Telefon 0 85 71/44 10

8400 Regensburg

Zimmermann elektroland
8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41/9 50 85
8390 Passau
Kohlbruck 2a
☎ 08 51/5 20 07

8423 Abensberg

COMPUTERVERSAND
WITTICH
Tulpenstr. 16 · 8423 Abensberg
☎ 0 94 43/453



8500 Nürnberg

EINE IDEE ANDERS
KARSTADT NÜRNBERG AN DER LORENZKIRCHE
TECHNIK CENTER
1. KLASSE EINKAUFEN IM WELTSTADTHAUS



8520 Erlangen



8520 Erlangen

Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31/4 20 76

Zimmermann elektroland
8520 Erlangen
Nürnberger Straße 88
Tel. (0 91 31) 3 45 68
8500 Nürnberg
Hauptmarkt 17
Tel. (09 11) 2 07 98

8600 Bamberg



8700 Würzburg

SCHOLL BÜROTEAM
Hardware · Software
Service · Schulung
computer center
am Dominikanerplatz
Ruf (09 31) 5 04 88

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21/65 21 54

8900 Augsburg



Unser Plus: Beratung u. Service
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (08 21) 52 85 33 oder 52 80 87
Computer Vertriebs- und Software GmbH

8900 Augsburg

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg · Tel. (08 21) 31 53-416

8940 Memmingen

EDV-Organisation
Hard- + Software
Manfred Schweizer KG
Ulmer Str. 2, Tel. 08331/12220
8940 Memmingen

Österreich

A-1020 Wien

W + H.
Computerhandel Ges m.b.H.

Förstergasse 6/3/2 · 1020 Wien
Tel. 0222-35 09 68

A-1040 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m.b.H.

1040 Wien - Paniglgasse 18-20
Telefon 02 22-65 78 08, 65 88 93

A-8010 Graz



A-8010 GRAZ
Mandellstraße 23;
Tel.: (0316) / 70 28 40-0*, 70 28 93-0*
Tx.: 31 25 34 zupan a

Schweiz

CH-1205 Geneve

PIMENT ROUGE
INFORMATIQUE S.A.
8, RUE DES MARAICHERS
1205 GENEVE TEL. 022/28 56 24

CH-1700 Fribourg

FRIDAT SA INFORMATIQUE
ehem. Softy Hard's Computershop

VOTRE SPECIALISTE
Rte des Grives 4
1700 Granges-Paccot/Fribourg
Tel. 0041 (0)37 26 66 28
Fax. 0041 (0)37 26 61 06

CH-2503 Biel

UE
URWA Electronic

Ihr ATARI ST Spezialist in
der Schweiz.
0 32 / 25 45 53

Lindenweg 24, 2503 Biel

CH-3084 Bern

Computer Corner
Ihr ATARI ST-Partner in Bern

Hard-Software / Beratung
Midi-Schulung / Service

Seftigenstr. 240 3084 Bern-Wabern
Zentrum Wabern Tel. 031/54 51 00

CH-4054 Basel

Wir sind Amiga- und ST-Freaks.
Unsere Öffnungszeiten:
Von Dienstag bis Freitag
9.30 – 12.30 und 14.00 – 18.30 und
am Samstag 9.30 – 16.00

SYSAG
COMPUTER CENTER
Basel Tel 061/39 25 25 · Holeestrasse 87 · 4054 Basel
Aarau Tel 064/22 63 33 · Kasernenstrasse 26 · 5000 Aarau

CH-4625 Oberbuchsiten

STECTRONIC M. Steck
Electronic-Computer-Shop

Hauptstr. 104/137
CH-4625 OBERBUCHSITEN
Tel. 062/63 17 27 + 63 10 27

CH-5430 Wettingen



Zentralstrasse 93
CH-5430 Wettingen
Tel. 056 / 27 16 60
Telex 814 193 seco

CH-8001 Zürich

ADAG LASERLADEN

SEILERGRABEN 41 . 8001 ZÜRICH
MO-FR: 12-18.30 TEL. 01/251 49 34

DTP... LASERPRINTS ... ET CETERA!

CH-8006 Zürich

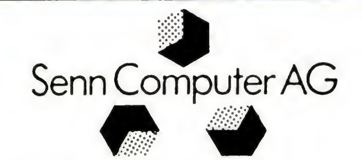
ADAG
COMPUTER-SHOP

UNIVERSITÄTSSTR. 25
8006 ZÜRICH TEL. 01/252 18 68

Computer-Center P. Fisch

Stampfenbachplatz 4
8006 ZÜRICH
01/363 67 67

CH-8021 Zürich



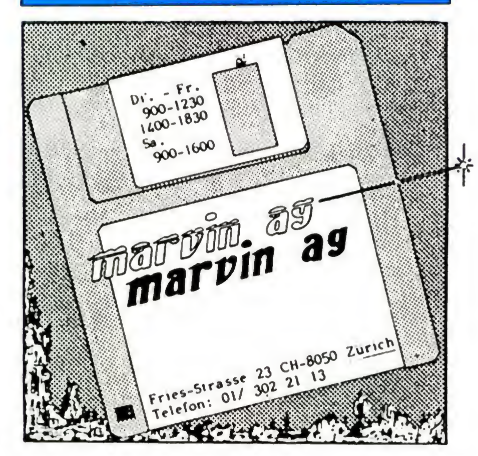
Langstrasse 31
Postfach
CH-8021 Zürich
Tel. 01/241 73 73
Telex 814 193 seco

CH-8021 Zürich

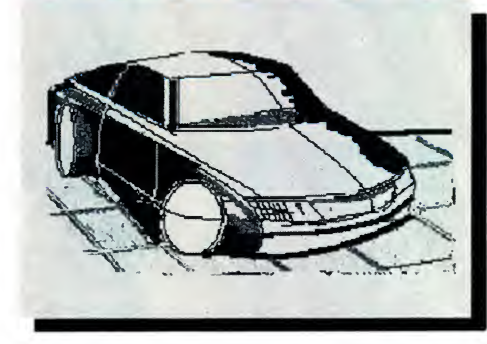
VILAN
Das Warenhaus der neuen Ideen

Ihr Computer-Fachhandel an der
Bahnhofstrasse 75 · Zürich

CH-8050 Zürich



CAMPUS ART



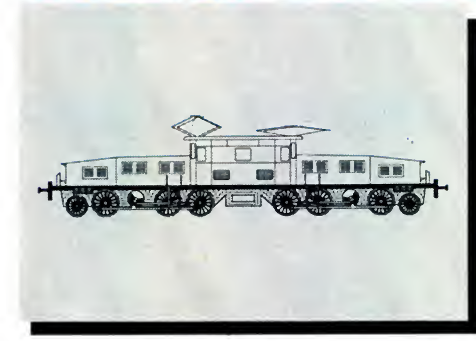
DM **149,-**

Das leistungsstarke Farbgraphiksystem für kreative Menschen!

- Arbeitstechniken wie im Graphicstudio (colorieren, schattieren, fotokopieren, ausschneiden, etc...)
- Eigenschaften der Stifte werden berücksichtigt
- Dreidimensionale Objektverarbeitung
- Benutzerfreundlichkeit durch Online Help

System: Atari ST, Farb-Bildschirm, Maus

CAMPUS DRAFT



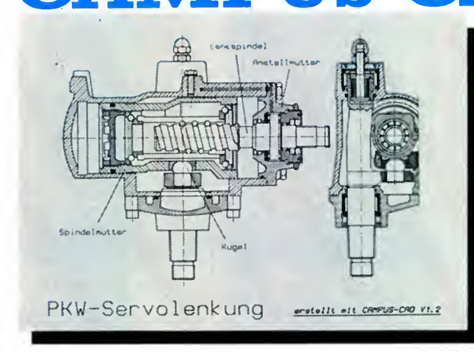
DM **149,-**

Vektororientiertes Zeichenprogramm mit allen Vorzügen professioneller CAD Systeme!

- Leichte Bedienung durch Maus, Menüs und Piktogramme
- Alle notwendigen Zeichenfunktionen (Lot, Parallele etc...)
- Vollständige Bemaßung, ausgereifte Symboltechnik
- Inclusive Drucker- und Plottertreiber

System: Atari ST, Monochrommonitor, Maus

CAMPUS CAD V1.



DM **798,-**

Neue Zusatzprogramme für das bewährte CAD-System!

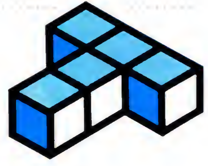
- **CAMPUS ASC:** Konvertiert CAD ↔ ASCII **248,– DM**
- **CAMPUS PLT:** Konvertiert HPGL → CAD **248,– DM**
- **CAMPUS SYM:** Symbolbibliotheken für Elektronik, Elektrotechnik, Hydraulik/Pneumatik u.a. **248,– DM**
- Universeller Plottertreiber

System: Atari 1040ST, Monochrommonitor, Maus

Technobox

Kornharpener Str.122a · 4630 Bochum 1
Telefon: 0234/503060 · FAX: 503061

CH: Senn Computer AG Zürich, Tel. 01/2428088
A: Steco Data, Dornbirn, Tel. 05572/65812



CH-8052 Zürich



CH-8200 Schaffhausen

ZIMELEC

CAR-AUDIO
COMPUTER + ELECTRONIC
Bachstraße 28 · 8200 Schaffhausen
Tel.: 053/55224

Montag—Freitag	9.00—12.00
	13.30—18.30
Samstag	9.00—16.00

CH-9001 St. Gallen

ADAG
COMPUTER-SHOP
TORSTR. 25
9001 ST.GALLEN
TEL. 071/25 43 42

CH-9400 Rorschach



Luxemburg



KFZ - Handel

Die
professionelle Lösung
für den Kraftfahrzeughändler

30 Minuten, das ist die Zeit, die Sie für die Einarbeitung in KFZ-Handel benötigen.

Nur 30 Minuten und ...
- KFZ-Handel verwaltet Ihre Fahrzeuge
- KFZ-Handel kalkuliert für Sie Aufwendungen, Umsatz, Preise, Steuern und Gewinne
- KFZ-Handel führt Feder bei Verträgen, Rechnungen, Quittungen und Agenturabrechnungen

KFZ-Handel auch dann, wenn perfekter Datenschutz gefordert ist, mehrere Betriebe geführt werden oder ... Rufen Sie uns an!

KFZ - Handel Softwarepaket kostet DM 1790.-

WDS Datensysteme

Danzigerstraße 9, 6094 Bischofsheim
Telefon 06144 - 4 15 05

Vertriebspartner gesucht



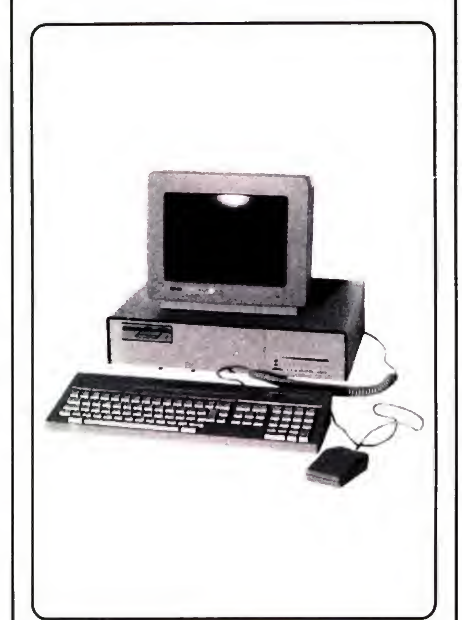
ATARI ST ALS PREISWERTES PROFISYSTEM

KOMPLETT-SYSTEME

Die berühmten Computer der ATARI ST-Serie, die modernste Technologien einsetzen, zu niedrigsten Preisen, sind jetzt als PROFESSIONELLES KOMPLETT-SYSTEM zu erhalten.
Das Herz dieses Systems, wie rechts abgebildet, ist der ATARI 1040 STF mit 1-MB-ARBEITSSPEICHER und einer modernen 3,5" DOPPELSEITIGER DISKDRIVE. Ein HOCHAUFLÖSENDER S/W MONITOR (SM 124), MAUS und BASIC machen die Grundversion komplett. Natürlich wird auch eine Version mit 20-MB-FESTPLATTE (SH 205) angeboten.
Die Vielfalt der PROFESSIONELLEN SOFTWARE, die nun für die ATARI-Rechner zur Verfügung steht, ist natürlich 100% lauffähig, und das flexible und geräumige Gehäuse erlaubt den Einbau von System-Erweiterungen.
Für den Kenner sind viele serienmäßige Extras eingebaut.

- Der Rechner wird beim Einschaltvorgang automatisch nach der Festplatte geordnet.
- Das sondergefertigte Hauptgehäuse verfügt über alle originalen Schnittstellen.
- Das Komplett-Gerät wird über einen Schalter geschaltet, die Festplatte kann jedoch bei Bedarf ausbleiben.
- Einbaumöglichkeit für ein weiteres 3,5"- oder 5,25"-Laufwerk und Harddisk bis zu 120 MB.
- FREIBEWEGLICHE TASTATUR mit Resetknopf und vieles mehr ...

L. H. 100 (System ohne Harddisk)	**1998,–**
L. H. 120 (mit 20-MB-Atari-Harddisk)	**3300,–**
L. H. 160 (mit 60-MB-Vortex Harddisk)	**4498,–**



BAUSÄTZE

beinhalten HAUPTGEHÄUSE, TASTATURGEHÄUSE und allen benötigten Platinen, Kabel und Kleinteile, um Ihren vorhandenen ATARI 260/520 oder 1040-Rechner, in das links beschriebene Profi-System umzubauen. Eine Umbauanleitung und technische Unterlagen sind beigefügt.
Der KOMPAKT-KIT 2 ist eine NEUENTWICKLUNG und ersetzt seinen populären Vorgänger, den die Atari-Fans seit 1½ Jahren erfolgreich einsetzen. Zu dem Umbau sind KEINERLEI LÖTARBEITEN erforderlich und er ist auch mit geringsten technischen Kenntnissen schnell und problemlos durchzuführen.
Alle HARDDISKS der Firmen ATARI und VORTEX sind ohne Zusatzteile einzubauen und unsere ZEITVERZÖGERUNG ist in jedem Kit serienmäßig dabei. Dazu passen alle gängigen 3,5"-LAUFWERKE, es wird sogar eine Blende für ein 5,25"-Laufwerk beigefügt. Bei dem KK2 260/520 wird ein SCHALTNETZTEIL mitgeliefert, dieses ersetzt das vorherige Gewirr von Netzteilen für Rechner und Floppies und erlaubt die Versorgung des kompletten Systems (außer Monitor) über ein zentrales Netzkabel und einen Schalter.
Unser FLACHES ABGESETZTES TASTATURGEHÄUSE, mit RESETKNOPF, voll entstörter Schnittstellenplatine und SPIRALKABEL, ist auch einzeln zu erhalten.

KK2 260/520	**498,—**
KK2 1040	**398,—**
Tastaturgehäuse für 260/520	**128,—**
Tastaturgehäuse	**128,—**

Bestellen Sie sofort oder fordern Sie Informationen an · Bei unserem VERSAND oder bei jedem guten FACHHANDEL · OEM-Anfragen erwünscht.



EINFÜHRUNGS-ANGEBOT BEI DIREKT-BESTELLUNG:

Komplettsystem LH 120 beinhaltet:

- Atari 1040 STF, Maus u. Basic mit
- doppelseitigem Diskettenlaufwerk 3½"
- Kompaktkitgehäusesystem
- Atari SM 124 Monitor
- Atari SH 205 Harddisk

2998 DM

Versand-Anschrift
Riedstraße 2
7100 Heilbronn
Telefon 07131/78480
Telefax 07131/79778

Kleinanzeigen

BIETE HARDWARE

OKI Microline 182 incl. Druck-Anpassung für 1st_Word+, Orig. Handbuch, Ersatzfarbband: DM 500,– Tel.: 06251/76703

Drucker Panasonic KX-P1092 G fast neuwertig – Originalverpackt DM 390,– Telephon: 089-8341503 ab 19.00 h

SF 354-Floppy 150,– 06269/759

Drucker Brother M-1509, DIN A3 DM 800,– Tel. 0911-560511

Citizen 120D. Tel. 07721-59937

SF354 Tel. 0541-596539

Durchsichtig! 260 ST im Plexigehäuse: 1MB + SF314 + 5 1/4-Teac (MS-DOS-fähig) + alt. u. neu-TOS umschaltbar + c't-Uhr + Schaltnetzteil + SM124 + Soft. VB 1900 069/618370

Systemwechsel (berufl.) SF 314 + NEC P6 + Traktor (3 Mon.) FP 1300 DM. Org. Softw. NP 5000 DM VB 1600 DM, bei Kauf viele Bücher + Zeits. gratis. Rufen Sie an, abends 0221/372793

SF314, VHP 350 DM, Abh., 06353/2419

SF354 100,– Druckerkabel 25,– NEC 1036A 175,– Tel: 0851/44415

Computer-Electronic-Zubehör Computer, ST-Tastaturgehäuse, Laufwerke, Floppy's, Disketten, Adapter a. Art, CBM-Ersatzchips, Software und vieles mehr bei: I. Herges, Obere Rischbachstr. 88 Tel. 06894/383178 6670 St. Ingbert

5 1/4" Fl-Station, neuw.+Garantie anschlf. VB 300,– Tel: 07191/53689

Drucker OKI-ML-182 incl. Kabel Traktor + Einzelblattzuführung VB 700 DM. Tel: 0471/33921

Monitor Sml24, ungebraucht, originalverpackt 300 DM Tel.: 05534/768

SF354 DM 150, ca. 1 Jahr, kaum gebraucht, Th. Resch, 02103-55148

■■■■■ Festplatten ■■■■■
■ für ATARI ST nach c't 2.88 ■
■Platten von 10-80MB; zB: 10MB■
■598DM,20MB 880DM,+Versand
■C-mos Ram f. Easyprommer 139,-
■Contec Inh.A.Niemann 1 Bln.44
■Mainzer Str. 3 Tel: 030/6227229

MEGA ST2, SH 205, LASER SLM804 günstig Tel. 08139/6444

James Seite 141

HDVORTEX 20+ 890,– 09131/990754

2xSF354. DM 200,– T. 0911/268379

○○○ TOP-ANGEBOT ○○○
Brenne TOS+BlitterTOS – billig! Auch beide Systeme zusammen im ST! Kinderleichter Einbau! Gratisinfo: M. Meyer / G. Rohlfs-Str. 54c / 2820 Bremen 70

BIETE SOFTWARE

Von Lehrer für Lehrer! Notenverw.prg Markbook plus in GEM. Über 20 versch. Funkt. f. alle gäng. Notensyst. Spez. vers. auf Wunsch; Info gegen Freiumschlag. H. J. Merkel/Nahestr. 28/6600 SBR

Von Bauleiter für Bauleiter!! Aufmass-Massen-Abrechnungs-Prg f. Atari-ST 245,- T. 0821-499518

Wärmebedarf DIN4701 + K-Zahl ★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder Eichendorffstr. 15 · 5030 Hürth

★Atari Lernkurs für Anfänger★
★Disk, Info: B. Biffiger, Dammw.25
★CH-3904 Naters (Rückporto)★

ATARIE ST-LOTTOSTATISTIK 39 DM, 20 seit. Dokumentation vorab gegen 2,40 in Briefmark. F. Klein, Zur Fröhn 15, 6601 Heusweiler, Tel. 06806/8901

■■■■■■■■■■■■■■■■■■■■
■★ Atari ★ Schulmeister ST■
■ Die komfortable Noten- u. ■
■ Klassenverwaltung mit ca. 40■
■Menüpunkten/GEM. Editier-■
■ bare Masken und Parameter:■
■ (Fächer, Unterbereiche, Ge- ■
■wichtung, Notentyp, Sortierkri-■
■ terien ect.) 500 Kbyte Ram ■
■monochrom-Monitor. Version■
■für BRD und SCHWEIZ. Aus-■
■ führl. Info/Freiumschlag. ■
■ M. Heber-Knobloch ■
■ Auf der Stelle 27 ■
■ 7030 Sindelfingen ■
■■■■■■■■■■■■■■■■■■■■

Public-Domain-Blitzversand!
Riesenauswahl! Preise:
ab DM 4,– einseitige Disketten
ab DM 6,– doppelseitige
Inclusive Diskette!!!
Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar!
Gratisliste anfordern bei:
A. Gauger Software
Buhlstraße 16a, 7505 Ettlingen
0 72 43 / 3 18 28
Bitte Computertyp angeben!!

James Seite 141

★Super-Lohn-Einkommensteuer Jahresausgleich 87 neu m. Datenspeicher, Kundenverw. Formulardruck, Analyse ab 70,– jährl. Aktu (10,–) Info gg. RP PC-Demodisk 10,– H-I-Software H-Ilchen Niederfelderstr. 44, 8072 Manching 08459-1669

○○○ TOP-ANGEBOT ○○○
1. SUPER PD-SOFTWARE...
2. Lernsystem, GEM, viele Modi
3. GfA-Basic Shell 4. Speedchip
Gratisinfo: M. Meyer / G. Rohlfs-Str. 54c / 2820 Bremen 70

STAR NL-10 Utility für Atari ST Kursivschrift in NLQ und Draft: Accessory m. Druckeranpassungen Handbuch & Disk: DM 30,– 040/868779

●ST-Guitarman/ST-Keyboardman Beide Programme stellen Musikakkorde grafisch dar. Nur 38,- DM. Info bei: A. Labermaier Bozener Str. 34, 8200 Rosenheim

PD-Grafik-Bilder, Art-Lib für Degas + Stad Katalog: Frey, Rheinstr. 12A 6538 Münster-Sarmsheim

8888888888888888888888888888888
8 Acht Zeilen pro Artikel! 8
8 das bietet Ihnen HAROFAKT! 8
8 DEMO 20,– Tel. 07346/3845 8
8888888888888888888888888888888

■Schweiz■ DISKETTEN■
3 1/2" No-Name DS/DD
ab 10 Stk FR 2,40/STK
3 1/2" Marke Brown SS/DD
ab 10 Stk FR 2,10/STK
Preise inkl. Versandkosten.
Nur solange Vorrat. Bestellen bei: Thomas Brendler, CH-7403 Rhäzüns

★★★ HANDWERKER ★★★
★HAROFAKT ist die perfekte★
★Fakturierung für Handwerker★
★ Kinderleicht! Mausgesteuert ★
★DM 698,– DEMO: 20,– DM★
★ Telefon: 07346/3845 ★

– Assembler 6502 – Programm für Commodore und Atari ST, incl. Einführung in Soft- und Hardware, Info 1,– R. Rosin, Peter-Spahn-Str. 4 · 6227 Winkel

PD-MB: 0571/710141 · 300/8/N/1

Wärmebedarf DIN4701 + K-Zahl ★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder, Eichendorffstr. 15 · 5030 Hürth

★Public-Domain-Kopierservice★ Wieder NEUE PD-Programme eingetroffen ★ Alle PD's dieser Ausgabe ★ einseitig 5,50 ★ zweiseitig 6,50 ★ Info kostenlos ★ Tel. 02954-1050

PD-Software ab 2,– DM/Diskette Angebot wie PD dieser Ausgabe Tel. 02721/2432

★★★★ PD-SERVICE ★★★★
Gratisinfo anfordern bei:
Roland Auditz · Braunschweiger Str. 21 · 3340 WF-Salzdahlum

STatiST – das modulare Statistik Programmpaket für den Atari ST z.Z. 16 Tests ★ Gem ★ Updates ★ Info anfordern (frank. Rückumschl.) ★ DM 198,– ★ Dr. K. Fröhlich, Wormser Str. 18, 6054 Rodgau 3

■■■■■PD ab DM 3,50■■■■■
■ PD 1 bis PD dieser Ausgabe ■
■ INFO-Liste gegen DM 0,80 ■
■ G. Grasberger, C-Orff-Str. 9 ■
■■■■8221 Traunwalchen■■■■

Flightsimulator II für ATARI ST DM 120,– 02206/6456

■■ Brandneue PD-Software■■
■Fast täglich neue Programme,■
■direkt vom Autor, die es nur■
■ bei mir gibt! Riesiges Ange- ■
■ bot auf über 30 Seiten, u. a. ■
■ alle PD's dieser Zeitschrift ■
■Programme sind einzeln aus-■
■ wählbar!!! Der absolute Hit:■
■■NUR 1,5 PF PRO KBYTE■■
■ Sie können eigene Disketten■
■senden oder auf Markendisks■
■ bestellen!! Jedem PD-Katalog■
■ liegt ein Extrablatt mit den ■
■allerneusten Programmen, die■
■ ständig erfaßt werden, bei!! ■
■ Bearbeitung erfolgt selbst- ■
■ verständlich postwendend!!! ■
■ Jetzt wieder ein erneuerter ■
GRATISkatalog bei Arne Zingel, Vermehrenring 11a, 24 Lübeck 1

PLOT_ST für Pro-FORTRAN! Einfachste Ausgabe grafischer Darstellungen auf Bildschirm und Drucker. Mehrere Funktionen gleichzeitig darstellbar. Kompatibel mit Großrechner Standard (Plot-79). DM 95,–. Handbuch DM 15,– (wird verrechnet); Info gratis. M. Gamer, Friedrichsring 26, 6050 Offenbach

James Seite 141

★STatistik ★ Analyse wiss. Daten f. ST ★ Lin. Nichtlin. Regression ★ alle übl. stat Tests ★ Info geg. Rückumschlag ★ Dr. Th. Rupprecht ★ 852 Erlangen Mozartstr. 63 (159,–)

PD Service Nord DS Disk ab 2,– Paket z. B. 81-90 auf 5 DS Disk nur 22,50 DM Angebot wie PD dieser Ausgabe Tel. 0431/94427

HEIM FIBU-ST! Die Fibu für den ST nur 125 DM, Demo 20 DM-Schein bei: MTC, Postfach 56 · A-6027 Innsbruck

★★★★ PD-SERVICE ★★★★
★ PD-Angebot dieser Ausgabe ★
★ Einzeldiskette SS DM 5,00 ★
★Doppeldiskette DS z. B. 1+2★
★33+34, 89+90 usw. DM 7,00★
★ Pakete 1–10, 11–20, usw. ★
★ auf je 10 Disk. DM 40,00 ★
★ auf je 5 D. (DS) DM 30,00 ★
★ Porto u. Verpack. DM 3,50 ★
★V.-Scheck o. NN (+DM 3,50)★
★N. Twardoch, Gröchteweg 22★
★ 4902 Bad Salzuflen 1 ★

★★KFZ-ST★★ Berechnung u. grafische Dokum. aller KFZ-Kosten unter GEM. Jeder Drucker anschließb. Disk 59 DM, Demo 10 DM Dipl. Ing. M. Heydrich, Lange Zeile 84 · 8520 Erlangen

Mainstream – ST – Mainstream User-Club für Atari-ST-Anwend. Info gegen Rückporto von: Kay-Uwe Berghof Roseggerstraße 5 5600 Wuppertal 2

Platinen-Layouts! von Elektroniksch., Atari-ST Plotterservice f. alle CAD Progr., 0431-180975 U. Tams, Westring 273, 2300 Kiel

ST-PD Disketten dieser Ausgabe /eigene PD ab 2 DM, 10er Pakete ab 35 DM, INFO: Witaseck, Josef-Neuberger-Str. 26, 4000 Düsseldorf

■ Alice Pascal-Interpreter ■ Orig., 150 DM, Tel. 06021/460101

Lohnsteuerjahresausgleich/Einkommensteuer 87/88. Grunddaten + Ergebnis speichern /WasWäreWenn 30 DM Vork. J. Höfer, Grunewald 2a, 5272 Wipperfürth 02192/3368

James Seite 141

Becker-Text DM 100,- 0821-706819

★ I have what you search !! ★
Tel.: 030/7820545
The Sunnyboys/TSB !!

Orig/Textomat-50-, Datamat-50-K-Spread 2-120-, K-Graph 2-70-TLDU-70-, Side Click-70-, Kalkum AT-100- Tel. 09131/31653

Deluxe Vokabelprogr. unter GEM mit 20000 engl. Vok., 5 Lernmodi, schnelles Laden. 79,– DM Info, Best.: Wolf, 0261-409959

Flugsimulator II ausführliche deutsche Übersetzung DM 20,– od. Verr.-Scheck an Pf 1365, 7312 Kirchheim

Fibu-MAN m, Orig. m. Handbuch zu verk. Tel. 07305/22174

□ □ SUCHEN SIE PD-SOFT?
□ □ Dann fordern Sie
□□ unseren Gratiskatalog
□ □
□ □ Klaus Kohler
Don-Carlos-Str. 33B
7 Stuttgart 80

★ Public-Domain-Kopierservice ★ Gratisinfo ★ Disketten ab 4 DM ★ M. Schönfelder ★ Tlf. 02954-1050

GROSSE GRAFIKSAMMLUNG! ca. 2000 IMG-Grafiken (z. B. für Wordplus) DM 20,– 06302-3338

★★ PD-Aktuell, das Infomagazin für den PD-Anwender. Probeheft bei EUSAX PD 4018 Langenfeld Haus Gravener Str. 85 T. 81232

Epson LQ-850 Druckertreiber f. WORDPLUS. Proport.lschrift im Blocksatz, IBM+Eps. Zeichensatz. verschied. Zeilenabstände, v. zusätzl. Steuerbef. HEX+CFG, uvam. 42,– incl. Anl. 02323/46329 K. Plüher, 4690 Herne 1, K.-F.-Friesenstr. 26 – Demo? –

Kassenbuch + Banküberweisung je 49 DM B. Eichhorn T. 030/667277

PD aus „ST" nur 4,80 DM incl. Disk. Ab Nr. 1. 24-h-Bestell-Service. Schneider 0281/61772

■■ PD-Express - J. Rangnow ■■
■
■■Public-Domain ohne Müll!!■
■■ Anzeige jetzt an anderer ■
■■ Stelle dieser Ausgabe ■
Anschauen, Staunen – ■
■
■■■PD-Express – Versand■■■

Original Publish.Partner u. BTX-Manager Tel. 08139/6444

ST-MAIL – DAS Mailboxprogr. Referenzliste / Info anfordern bei M. Nitschke Softw.-Entwicklung u. -Vertrieb Bahnhofstraße 21 · 6330 Wetzlar

LERNPROGRAMME
für den ATARI ST
1. Kommerzielle ST-Lernprogr.
2. PD-Lernprogramme für den ST
3. PD-Lernprogramme für den PC
Katalog anfordern bei:
LTA – LERNSOFT
Friedenstr. 7 · 8032 Lochham
089/8545531

Minerale-Prg. m. Druckerausgabe von Achat bis Zoisit über 250 Minerale. Best. mit Überweisung v. 25,–; George Pskto 90203-107 PGA Bln. West 10010010. Abs. nicht vergessen! Prg a. 1DD, Verp.-Versandk. incl.

SUCHE HARDWARE

Suche NEC-P6. Tel: 0641/34533

SUCHE SOFTWARE

SKAT mit Null + Ramsch gesucht Hendricks, Oberwinzerfeld 27, 4320 Hattingen, Tel. 02324/81408

Suche ST-Software: Liste an – Patricia Purdy Bleichstr. 19 7562 Gernsbach Danke!

ST-Soft-Tausch! Tel: 04292/1669

Su. Wordpl. Treiber f. Citizen 120D sowie Landkarten Mitteleuropa. 0551-77370 M. Bartmann

TAUSCH

Ich digitalisiere Eure Fotos Tausche auch PD-Soft (Grafik usw.) Tel. bei Frank 05241/28015

FRIVOLE SOFTWARE. Liste gegen 50 Pf. Rückumschlag. Tausche keine Raubkopien. Fischer, Postfach 1619 4830 Gütersloh

KONTAKTE

! 3-D PROGRAMMIERER Team für 3-D Programmierung aus ASCII-Schnittstelle ges. Architektur-EDV Chiffre ST-588

★★ST-Softwareentwicklung★★ Schreibe Software, die speziell auf Ihre Bedürfnisse zugeschnitten ist. Ksl. Info anford. Thomas Leufkes, Knappenstr. 10, 4353 Oer-Erkenschwick, 02368/55020

VERSCHIEDENES

Übernehme Programmieraufgaben in GFA-BSIC P. Neurohr Eschersheimer Landstr. 273 6 Frankfurt 1 Tel. 069/567298

★ NEC-1037 Diskstationen ★
★ Einzel- und Doppellaufw. ★
★ Vortex Festpl. Billiger ★
★ als anderswo. Verhandelt ★
★ mit uns. Software billig ★
★ – Liste STAD, Signum u.a. ★
★ STEVE 3.0 mit Handyscan ★
★ATARIPOINT 07031/52607★

FLUGSIMULATOR II f. Atari ST ausführl. deutsche Übersetzung. INFO Pf 1365, 7312 KIRCHHEIM

James Seite 141

Suche Kontakte zu Programmierfreaks die Lust und Energie haben den Atari ST aus seiner Spieler-Ecke zu holen. Motto: es steckt viel drin, holen wir es raus. Entwicklung anwenderfreundlicher Softwaremodule für Dienstleistungsbranchen. Stuttgart 0711/465485.

ST-Ecke

AUF BÄUME GEKLETTERT...

Schon vor einigen Ausgaben beschäftigte ich mich mit der Baumstruktur der Dialogboxen. Oft stellt sich dabei das Problem, daß in einer Baumstruktur eine Vielzahl von Objekten bearbeitet werden muß. Eine einfache aber unsaubere Möglichkeit besteht darin, den Baum von Objekt 0 bis zum letzten Objekt hintereinander 'abzugrasen', aber dazu muß man die Baumlänge kennen. Viel besser und auch immer anwendbar ist das strukturierte Bearbeiten der Baumstruktur. Wie man ein solches Bearbeiten realisiert, soll diesen Monat an der Routine work_tree() gezeigt werden.

Einigen Lesern wird der Aufbau einer Objektstruktur bekannt sein, trotzdem möchte ich noch einmal zum besseren Verständnis der vorgestellten Routine kurz darauf eingehen. Jedes Objekt in einer Dialogbox oder einer Menüleiste besitzt einen Eintrag in einer Liste, der wiederum verschiedene Informationen enthält. Diese Einträge haben folgende Namen:

```
int           ob_next;
int           ob_head;
int           ob_tail;
unsigned int ob_type;
unsigned int ob_flags;
unsigned int ob_state;
char         *ob_spec;
int           ob_x;
int           ob_y;
int           ob_width;
int           ob_height;
```

Das bedeutet also, daß ein Objektbaum aus einer Wiederholung dieser Objektstruktur besteht. Für unser heutiges Thema interessieren uns nur die ersten drei Einträge der Objektstruktur, die eine Verkettung der einzelnen Objekte ermöglicht. Weitere ausführliche Informationen über die Objektstruktur finden Sie in unserem 2. ST-Computer-Sonderheft. Stellen wir uns als Dialogbox folgendes Beispiel vor: Eine große Box enthält fünf kleinere Boxen, in denen jeweils fünf Texte vorhanden sind, was bedeutet, daß die einzelnen Objekte verschachtelt sind. Die große Box (Objekt Nummer 0) enthält fünf kleinere Objekte (auf gleicher Ebene), deren wiederum jedes fünf kleinere Objekte enthält. Der Eintrag ob_next zeigt auf den Feldindex der auf gleicher Ebene befindlichen Objekte. So zeigt also die kleine Box 1 auf die nächste Box 2, wobei diese wiederum auf Box 3 zeigt. Ist die Objektliste beendet, findet sich in diesem Eintrag eine -1. Bitte betrachten Sie nun die Liste bau[] im Listing. Sie erkennen, daß der erste Eintrag des ersten Objektes eine -1 besitzt, was bedeutet, daß ein weiteres Fortfahren an dieser Stelle nicht gegeben ist, da die weiteren Objekte der Dialogbox sich nicht mehr auf gleicher Ebene befinden. Die Verschachtelung ergibt sich nun durch die Einträge ob_head (Kopf der Verschachtelung) und ob_tail (Schwanz der Verschachtelung). In unserem Listing zeigt nun ob_head auf das Objekt 1 und ob_tail auf Objekt 5. Das bedeutet, daß das erste Objekt, das sich in der großen Box befindet, Objekt 1 und das letzte Objekt der großen Box Objekt 5 ist. Schaut man sich nun den Eintrag ob_next von Objekt 1 an, so sieht man, daß dieser auf Objekt 2, ob_next von diesem auf Objekt 3, ob_next von diesem auf Objekt 4, ob_next von diesem auf Objekt 5 und ob_next von diesem wieder auf die große Box Objekt 0 zeigt. Es wäre aber falsch anzunehmen, daß diese serielle Verknüpfung immer so sein muß, vielmehr könnte die Verzeigerung völlig durcheinander sein. In unserem Beispiel sehen wir, daß die Objekte 1 bis 5 wiederum in Untergruppen verzweigen. Ist keine Verzweigung mehr vorhanden, so steht in ob_head und ob_tail eine -1.

Welche Arbeit muß nun unsere Routine durchführen ? Zunächst schreiben wir uns eine Routine, die sich durch die Verzeigerung von ob_next von einem Objekt zum anderen hangelt. Dies könnte folgendermaßen realisiert werden:

```
Objektnummer =0
Solange
   Objekt bearbeiten...
   Objektnummer = ob_next-
     Eintrag des aktuellen Ob-
     jektes
wie ob_next-Eintrag ungleich -1
```

Dadurch können Objekte in einer bestimmten Ebene bearbeitet werden, allerdings würden dadurch alle Verzweigungen außer acht gelassen. Den Anfang und das Ende einer Verzweigung können wir uns aber aus einem Objekt in ob_head und ob_tail herauslesen. Ist keine Verzweigung vorhanden, so erkennen wir das an einer -1 in ob_head und ob_tail. Vervollständigen wir unsere Schleife also folgendermaßen:

```
Objektnummer =0
Solange
  Objekt bearbeiten...
  Ist ob_head und ob_tail
   ungleich -1
    dann ist Verzweigung vor-
    handen: Bearbeite Ver-
    zweigung ob_head bis
    ob_tail
  Objektnummer = ob_next-
   Eintrag des aktuellen Ob-
   jektes
wie ob_next-Eintrag ungleich -1
```

Wie muß nun die Bearbeitungsroutine dieser Verzweigung aussehen ? Richtig ! Im Prinzip genauso wie die, die wir gerade entwickelt haben. Sie muß nur als Abbruchbedingung das Erreichen des Objektes mit der Objektnummer ob_tail haben. Fügen wir also diese Abbruchbedingung ein:

```
Objektnummer =0
Solange
   Objekt bearbeiten...
   Ist ob_head und ob_tail un-
    gleich -1
   dann ist Verzweigung vor-
    handen: Bearbeite Verzwei-
    gung ob_head bis ob_tail
   Ist Objektnummer gleich mit
    ob_tail
   dann breche Schleife ab
   Objektnummer = ob_next-
    Eintrag des aktuellen
    Objektes
wie ob_next-Eintrag ungleich -1
```

Bemerken Sie etwas ? Die Routine, die ursprünglich nur eine Ebene bearbeiten konnte, kann nun auch tiefere Ebenen bearbeiten. Was also liegt bei einer Verzweigung näher, die gleiche Routine wieder aufzurufen ?! Dazu setzen wir die Objektnummer nicht auf 0, sondern übergeben diesen Wert der Routine als Anfangswert, so daß die Routine allgemein geschrieben folgendermaßen aussieht:

```
WORK_TREE
Parameter: Baumadresse, An-
fangsobjekt, Endobjekt, Routi-
nenname

Objektnummer = Anfangsobjekt
Solange
   Routine 'Routinenname' auf-
    rufen

   Ist ob_head und ob_tail un-
    gleich -1
      dann ist Verzweigung
       vorhanden: WORK_TREE
       mit ob_head und ob_tail
       aufrufen
   Ist Objektnummer gleich mit
    ob_tail
      dann breche Schleife ab

   Objektnummer = ob_next-
    Eintrag des aktuellen Ob-
    jektes
wie ob_next-Eintrag ungleich -1
```

Je nachdem wie tief die Verschachtelung des Baumes ist, kann es schon sein, daß work_tree (so der Name unserer Routine) sich mehrfach selbst

```
 1:  #include <aes_vdi.h>
 2:
 3:  /* Text-Strings der Objekte */
 4:
 5:  char *strings[] = {
 6:  "Objekt 6",
 7:  "Objekt 7",
 8:  "Objekt 8",
 9:  "Objekt 9",
10:  "Objekt 10",
11:  "Objekt 11",
12:  "Objekt 12",
13:  "Objekt 13",
14:  "Objekt 14",
15:  "",
16:  "Objekt 15",
17:  "Objekt 16",
18:  "Objekt 17",
19:  "Objekt 18",
20:  "Objekt 19",
21:  "Objekt 20",
22:  "Objekt 21",
23:  "Objekt 22",
24:  "Objekt 23",
25:  "Objekt 24",
26:  "Objekt 25"
27:  };
28:
29:  /* Text-Informationen der Objekte */
30:
31:  TEDINFO ted[] = {
32:  strings[8], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
33:  strings[10], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
34:  strings[11], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
35:  strings[12], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
36:  strings[13], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
37:  strings[14], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
38:  strings[15], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
39:  strings[16], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
40:  strings[17], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
41:  strings[18], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
42:  strings[19], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1,
43:  strings[20], strings[9], strings[9], 3, 6, 0, 0x1180, 0x0, -1, 10,1};
44:
45:  /* Objektbaumstruktur */
46:
47:  OBJECT bau[] = {
48:  -1, 1, 5, G_BOX, NONE, OUTLINED, 0x21100L, 0,0, 624,288,
49:  2, 6, 9, G_BOX, NONE, NORMAL, 0xFF1100L, 16,64, 112,192,
50:  3, 10, 13, G_BOX, NONE, NORMAL, 0xFF1100L, 136,64, 112,192,
51:  4, 14, 17, G_BOX, NONE, NORMAL, 0xFF1100L, 256,64, 112,192,
52:  5, 18, 21, G_BOX, NONE, NORMAL, 0xFF1100L, 376,64, 112,192,
53:  0, 22, 25, G_BOX, NONE, NORMAL, 0xFF1100L, 496,64, 112,192,
54:  7,  -1, -1, G_STRING, NONE, NORMAL, strings[0], 24,16, 72,16,
55:  8,  -1, -1, G_STRING, NONE, NORMAL, strings[1], 24,64, 72,16,
56:  9,  -1, -1, G_STRING, NONE, NORMAL, strings[2], 24,112, 72,16,
57:  1,  -1, -1, G_STRING, NONE, NORMAL, strings[3], 24,160, 72,16,
58:  11, -1, -1, G_BUTTON, 0x40, NORMAL, strings[4], 24,16, 72,16,
59:  12, -1, -1, G_BUTTON, 0x40, NORMAL, strings[5], 24,64, 72,16,
60:  13, -1, -1, G_BUTTON, 0x05, NORMAL, strings[6], 24,112, 72,16,
61:  2,  -1, -1, G_BUTTON, 0x05, NORMAL, strings[7], 24,160, 72,16,
62:  15, -1, -1, G_BOXTEXT, NONE, NORMAL, &ted[0], 24,16, 72,16,
63:  16, -1, -1, G_BOXTEXT, NONE, NORMAL, &ted[1], 24,64, 72,16,
64:  17, -1, -1, G_BOXTEXT, NONE, NORMAL, &ted[2], 24,112, 72,16,
65:  3,  -1, -1, G_BOXTEXT, NONE, NORMAL, &ted[3], 24,160, 72,16,
66:  19, -1, -1, G_TEXT, NONE, NORMAL, &ted[4], 24,16, 72,16,
67:  20, -1, -1, G_TEXT, NONE, NORMAL, &ted[5], 24,64, 72,16,
68:  21, -1, -1, G_TEXT, NONE, NORMAL, &ted[6], 24,112,72,16,
69:  4,  -1, -1, G_TEXT, NONE, NORMAL, &ted[7], 24,160,72,16,
70:  23, -1, -1, G_BOXTEXT, NONE, SHADOWED, &ted[8], 24,16, 72,16,
71:  24, -1, -1, G_BOXTEXT, NONE, SHADOWED, &ted[9], 24,64, 72,16,
72:  25, -1, -1, G_BOXTEXT, NONE, SHADOWED, &ted[10], 24,112, 72,16,
73:  5,  -1, -1, G_BOXTEXT, LASTOB, SHADOWED, &ted[11], 24,160, 72,16};
74:
75:  extern selec_objc(), deselec_obj();     /* Vorwärtsdeklaration */
76:
77:  main()
78:  {
79:     int d;                               /* Dummy-Variable */
80:     appl_init();                         /* AES initialisieren */
81:
82:     form_center(bau, &d, &d, &d, &d);    /* Box zentrieren */
83:     work_tree(bau, 0,0, selec_objc);     /* Alle Objekte bearbeiten */
84:     printf("Alle Objekte selektiert !!!\n");
85:     work_tree(bau, 0,0, deselec_objc);   /* Alle Objekte bearbeiten */
86:     printf("Alle Objekte deselektiert !!!\n");
87:     gemdos(7);                           /* Auf Tastendruck warten */
88:
89:     appl_exit();
90:  }
91:
92:  selec_objc(tree, which)
93:  OBJECT *tree;
94:  int which;
95:  {
```

aufruft. Übrigens, ein solches Aufrufen der eigenen Routine nennt man rekursiv und wird gerade bei Baumstrukturen sehr oft angewendet. Es soll allerdings nicht verschwiegen werden, daß solche Routinen bei häufigen Selbstaufrufen kurzzeitig viel Speicherplatz verbrauchen. Dies ist in unserem Beispiel nicht so schlimm, da ein Objektbaum höchstens eine Verschachtelung von sieben Ebenen zuläßt. Eine Implementierung der Routine work_tree() in der Sprache 'C' habe ich als Listing 1 mitgeliefert. Durch die allgemeingehaltene Herleitung der Routine müßte es aber einfach sein, auch Implementierungen in anderen Sprachen durchzuführen. Implementierungen in einer anderen Sprache sind gern willkommen und würden in einer der nächsten ST-Ecken veröffentlicht. An dieser Stelle möchte ich die Frage an den Leser weitergeben, was er von solchen allgemeinen Darstellungen einzelner Routinen hält?!

Beim Aufruf der angegeben Routine werden übrigens als Parameter die Adresse der Baumstruktur und die Nummer des aktuellen Objektes übergeben. In meinem Beispiel werden damit erst alle Objekte selektiert und dann deselektiert. Es ist natürlich selbstverständlich, daß die entsprechende Routine eine entsprechende Parameterliste enthalten muß. Theoretisch wäre es auch möglich in der aufgerufenen Routine alle Textfelder zu löschen oder, oder, oder... In der Praxis sollte man übrigens noch ein paar zusätzliche Abbruchbedingungen einbauen: So sollten alle Untergruppen ausgelassen werden, bei denen das HIDDEN-Flag gesetzt worden ist. Außerdem sollte auf das Flag LASTOB in ob_flags abgefragt und falls gesetzt, die Schleife abgebrochen werden, was aber hier nicht so einfach ist, da dieses Objekt in einer der tieferen Ebenen liegen könnte.

In der Hoffnung, daß ab sofort alle Objektbearbeitungen auf diese Weise durchgeführt werden und nicht mehr von Objektnummer 0 bis zum Ende, verbleibe ich bis zum nächsten Monat

Stefan Höhn

```
 96:    tree[which].ob_state |= SELECTED;     /* Objekte selektieren */
 97:    objc_draw(tree,which,0,0,0,640,400); /* und zeichnen */
 98:    gemdos(7);                            /* Auf Tastendruck warten */
 99: }
100:
101: deselec_objc(tree, which)
102: OBJECT *tree;
103: int which;
104: {
105:    tree[which].ob_state &= ~SELECTED;    /* Objekte deselektieren */
106:    objc_draw(tree,which,0,0,0,640,400); /* und zeichnen */
107:    gemdos(7);                            /* Auf Tastendruck warten */
108: }
109:
110: /*********************************************************************/
111: /*                                                                   */
112: /* work_tree wandert die Objektstruktur eines Baums ab und führt dabei */
113: /* eine Routine aus, die das aktuelle Objekt bearbeitet. Dieser Routine */
114: /* werden die Baumadresse und der Indes des momentan aktuellen Objektes */
115: /* übergeben. 'Achtung': work_tree ruft sich rekursiv auf...         */
116: /*                                                                   */
117: /* Parameter: baum    -  Baumadresse                                 */
118: /*            ob_an   -  Anfang der Wanderung                        */
119: /*            ob_en   -  Ende der Wanderung                          */
120: /*            aufruf  -  Routine, die ausgeführt werden soll         */
121: /*                                                                   */
122: /* Soll der gesamte Baum bearbeitet werden, werden ob_an und ob_en auf */
123: /* Null gesetzt.                                                     */
124: /*                                                                   */
125: /*********************************************************************/
126:
127: work_tree(baum, ob_an, ob_en, aufruf)
128: OBJECT *baum;                         /* Adresse der Baumstruktur */
129: int ob_an, ob_en;                     /* Anfang und Ende */
130: int (*aufruf)();                      /* Routine, die aufgerufen wird */
131: {
132:    int ob_ind;                        /* aktueller Objektindex */
133:
134:    ob_ind= ob_an;            /* Index ist erstes Objekt */
135:
136:    do                                 /* 'Endlos' bis zum break */
137:    {
138:       (*aufruf)(baum,ob_ind);         /* Aufruf der Routine */
139:       if (baum[ob_ind].ob_head!=-1) /* falls Verzweigung vorhanden */
140:          work_tree(baum, baum[ob_ind].ob_head, baum[ob_ind].ob_tail, aufruf
             );
141:
142:       if (ob_ind==ob_en)             /* letztes Objekt bearbeitet */
143:          break;                       /* Schleifenende */
144:       ob_ind=baum[ob_ind].ob_next; /* nächstes Objekt */
145:    }while (baum[ob_ind].ob_next!=-1);
146: }
```



ENDE

INSERENTEN-VERZEICHNIS

ATARI MULTISYNC SM-124

Gibt es nicht? Gibt es doch! Mit unserer Umbauanleitung kann jeder versierte Hobbyelektroniker aus seinem Schwarz/Weiß-Monitor einen Monitor für alle drei Auflösungsstufen bauen. Nur für die Farbe brauchen Sie nach wie vor Pinsel und Farbenkasten.

Vorab müssen wir Sie aber auf ein Problem aufmerksam machen. In der Redaktion ist über die Veröffentlichung dieses Artikels eine wilde Diskussion entbrannt. Die Einen pochen auf die Verantwortung gegenüber der Leserschaft und wollen eine Umbauanleitung, die mit Sicherheit bei einigen Lesern zu Problemen führen wird (alleine schon dadurch begründet, daß wir im Laufe der Recherche vier verschiedene Monitorversionen gefunden haben - wer weiß, wieviel Versionen es insgesamt gibt...), unter diesen Voraussetzungen nicht veröffentlichen, und die Anderen, die dem Leser seine eigene Verantwortung nicht nehmen wollen - auch auf die Gefahr hin, daß einige Monitore und hoffentlich kein Leser seinen Geist aufgeben wird.

Nun, den Artikel haben Sie vor sich liegen - trotzdem möchten wir es klar betonen: Diese Umbauanleitung ist nichts für den Laien. Sie sollten über gewisse Grundkenntnisse in der Fernsehtechnik verfügen. Weiterhin sollte Ihnen ein kleiner Meßgerätepark zur Seite stehen, um bei Problemen zumindest den Urzustand wieder herzustellen. Dennoch ist die Gefahr sehr groß, daß Ihnen selbst das nicht mehr gelingt, da der ATARI-Monitor sehr "Bauteilekritisch" aufgebaut ist. Die größere Gefahr betrifft aber Ihre eigene Person. Abgesehen von den lebensgefährlichen Spannungen, die in dem Gerät "frei zugänglich" sind, besteht auch die Gefahr einer Implosion der Bildröhre (z.B. durch Umkippen des Monitors). Schützen Sie sich durch eine Brille vor Verletzungen der Augen und benutzen Sie beim Abgleich zumindest einen Trenntrafo (trotzdem bleiben die Spannungen lebensgefährlich!). Tragen Sie außerdem isolierende Schuhe.

Zum Schluß: Sollten Sie die leisesten Zweifel an Ihrer Umbaubefähigung haben - lassen Sie lieber die Finger davon.

Und wer es nun noch immer nicht lassen kann - hier die Umbauanleitung:

Bevor Sie richtig loslegen, stellen Sie erst einmal Ihren Monitor auf den Kopf und sehen Sie durch die Lüftungsschlitze. Sollten Sie drinnen eine dunkelgrüne Platine sehen, dürfen Sie sich freuen. Wenn Sie stattdessen ein graues Blech erblicken, bereitet das größere Probleme, denn eigentlich müßten hier viele Bauelemente und Kabel von unten angelötet werden.

Die neueren Monitore mit dem Blech auf der Platinenunterseite (ab etwa 6/1987) konnten nicht so ausführlich in der Anleitung behandelt werden. Sie müßten in diesem Fall alle Bauteile auf einer größeren Lochrasterplatte unterbringen und außerdem die Leiterplatte sehr mühsam ausbauen. Eventuell ist dann der Kauf eines Modulators für Sie doch interessanter.

Der Umbau wird jetzt Schritt für Schritt erklärt. Schauen Sie sich die Detailfotos an, auf ihnen haben wir die jeweiligen Positionen noch zusätzlich mit Pfeilen markiert. Außerdem können Sie die Orginalbauteile anhand ihrer Bezeichnung auf der Platine finden:

Vorab: Netzstecker aus der Steckdose!

1. Zuerst wird das zusätzliche Videokabel installiert:

1.1 Öffnen Sie den Monitorstecker und bauen Sie die Widerstände R1, R2 und R3 ein.
Dazu sind einige Steckstifte zu lösen. Bild 1 hilft beim späteren Zusammenbau.

Pin Nr.	*Funktion*	*Anschluß*
1	*Audio Ausgang*	*gelb*
4	*Monochrom Sensor*	*schwarz*
6	*Grün Ausgang*	*an R2*
7	*Rot Ausgang*	*an R1*
8	*12V Ausg. (10mA)*	*weiß*
9	*Horiz. Sync. Ausg.*	*blau*
10	*Blau Ausgang*	*an R3*
11	*Monochrom Signal Ausgang*	*braun*
12	*Vert. Sync. Ausg.*	*grün*
13	*Masse*	*grau*

Die übrigen Pins werden nicht gebraucht

Bild 1

Leider kann es auch hier ein Problem geben:

Wenn der Monochromsensor im Stecker selbst an Masse angeschlossen ist, muß diese Brücke durchgetrennt und ein weiteres Kabel von Pin 4 außen am Originalkabel entlang zum Monitor geführt werden.

1.2 Bohren Sie ein Loch (3mm Durchmesser) in die Steckerhülle und löten Sie die Ader des dünnen abgeschirmten Kabels mit den freien Enden von R1 - R3 zusammen.

Bohren Sie das Loch dicht neben die Knickschutztülle. Die Abschirmung vom dünnen Kabel braucht im Stecker nicht angeschlossen zu werden.

1.3 Der Monitorstecker wird wieder zusammengebaut. Die Kabel werden mit Kabelbindern in kurzen Abständen gebunden.

2. Es folgen alle mechanischen Arbeiten:

2.1 Entfernen Sie die Gehäuserückwand vom Monitor (5 Schrauben und 1 Stecker für Lautsprecher).

2.2 Bohren Sie ein Durchgangsloch (3.5mm Durchmesser) für das dünne Kabel 3mm oberhalb des Dicken.

2.3 Befestigen Sie den Drehschalter am Blech und bohren Sie ein Durchgangsloch an gleicher Stelle in die Gehäuserückwand.

Beim Bohren dürfen keine Metallspäne in das Gerät kommen. Am besten bedeckt man die Elektronik vorher mit einem Handtuch. Zum Löten muß der Schalter nachher einige Male gelockert und etwas gedreht werden. Vorsicht ist beim Bohren geboten. Die niedrigste Drehzahl ist unbedingt einzustellen, oder man benutzt eine Handbohrmaschine.

3. Der Analogverstärker wird aufgebaut (Lochraster ca. 15 x 30mm) und an den Video-Verstärker angelötet (Buchstabe K-M). (Bild 2).

Dazu muß die Video-Verstärker-Platine vorsichtig von der Bildröhre abgezogen werden.

4. Die nun folgenden Änderungen finden auf der Grundplatine statt. (Bild 3):

4.1 Ersetzen Sie R 606 (330 k) durch 470 k. *Der neue Widerstand kann selbstverständlich auch von unten angelötet werden.*

4.2 Ersetzen Sie Diode D708 (RGP 10 G) durch Diode RGP 30 G (von unten anlöten). *Die Diode sitzt gut versteckt zwischen Zeilentrafo und Kühlblech.*

4.3 Löten Sie von unten 2 Drahtbrücken: IC 302 Pin 8 - Pin 10 und IC 302 Pin 9 - Pin 11.

4.4 Bringen Sie auf der Platinenunterseite, wie im Schaltbild und Bild 3 angegeben (für Buchstabe A-C, I, N, O und Q), folgende Bauteile an:

- C2
- R9, C3, D3, R8
- R10
- C4
- R11, C7, C8, C9 *(Kürzen Sie die Beinchen vom Zeilentrafo, damit diese nicht einen Kondensator aufspießen können; sonst droht ein Kurzschluß mit gefährlicher Hochspannung).*

Falls die Platine ihres Monitors von unten nicht zugänglich ist, müssen Sie alle diese Bauteile auf eine Lochrasterplatine löten. (Ab ca. 6/87)

5. Verdrahten Sie mit Schaltlitze nach Schaltbild alles außer Antennenkabel und Buchstabe D-G. Diode D4 und die Drahtbrücken werden zuvor an den Schalter gelötet.

Die meisten Kabel werden locker von der Platinenunterseite zum Schalter geführt. Auf Kabelbinder sollte aus hochfrequenztechnischen Gründen verzichtet werden. Die beiden Schalterebenen sind in der Kontaktanordnung nach dem Schaltbild zu verdrahten.

Schieben Sie einen Isolierschlauch Über alle Bauteile.

Bei neuerem Monitor:
Viele Anschlüsse können auch von oben hergestellt werden. Aber wenn das Blech am neueren Monitor Ihnen die Sicht nimmt, ist es jetzt an der Zeit die Grund-Platine komplett auszubauen. Dazu müssen Sie seitlich 2 Schrauben lösen. Das Chassisblech könnte ebenfalls abgeschraubt werden (Siehe 7.)

6. Der Monochrom-Sensor wird angeschlossen (Buchst. F):

Ziehen Sie den Stecker des Monitorkabels (P301) von der Grundplatte und trennen Sie den Monochrom-Sensor-Anschluß ab (schwarzes Kabel, ehemals an P30

Das zum Computer führende Kabelende wird nach Schaltbild an D4 angeschlossen. Das zum Stecker führende Ende bleibt unbenutzt.

Wenn sie unter 1.1 das Zusatzkabel anschließen mußten, wird dieses Kabel zu D4 geführt. P301 bleibt unverändert.

7. Jetzt kommt der schwierigste Teil (weil kaum zugänglich):

7.1 Stellen Sie den Monitor in Normallage auf den Tisch. Kippen Sie ihn nun auf die Bildschirmseite (weiche Unterlage benutzen) und lösen Sie die 3 Halteschrauben für das Chassis. Kippen Sie den Monitor wieder in Normallage und rücken Sie das Chassis vorsichtig vom Monitor ab. Ziehen Sie den Stecker, wenn erforderlich.

7.2 Trennen Sie die Anode von Diode D707 von der Leiterplatte und biegen Sie die Diode senkrecht nach oben. Ersetzen Sie C716 durch C5,C6,D5 nach Schaltbild *(auf Polarität der Diode achten)* und schließen Sie das flexible Antennenkabel an. *(Schirm=O) Anode von D5 und Buchstabe N an das freie Beinchen von Diode D707).*

7.3 Isolieren Sie das ganze gut, da sich der Netzschalter in direkter Nachbarschaft befindet (Lebensgefahr). Auch von der schwarzen Kohleschicht (leitfähig!) auf der Bildröhre droht Kurzschlußgefahr.

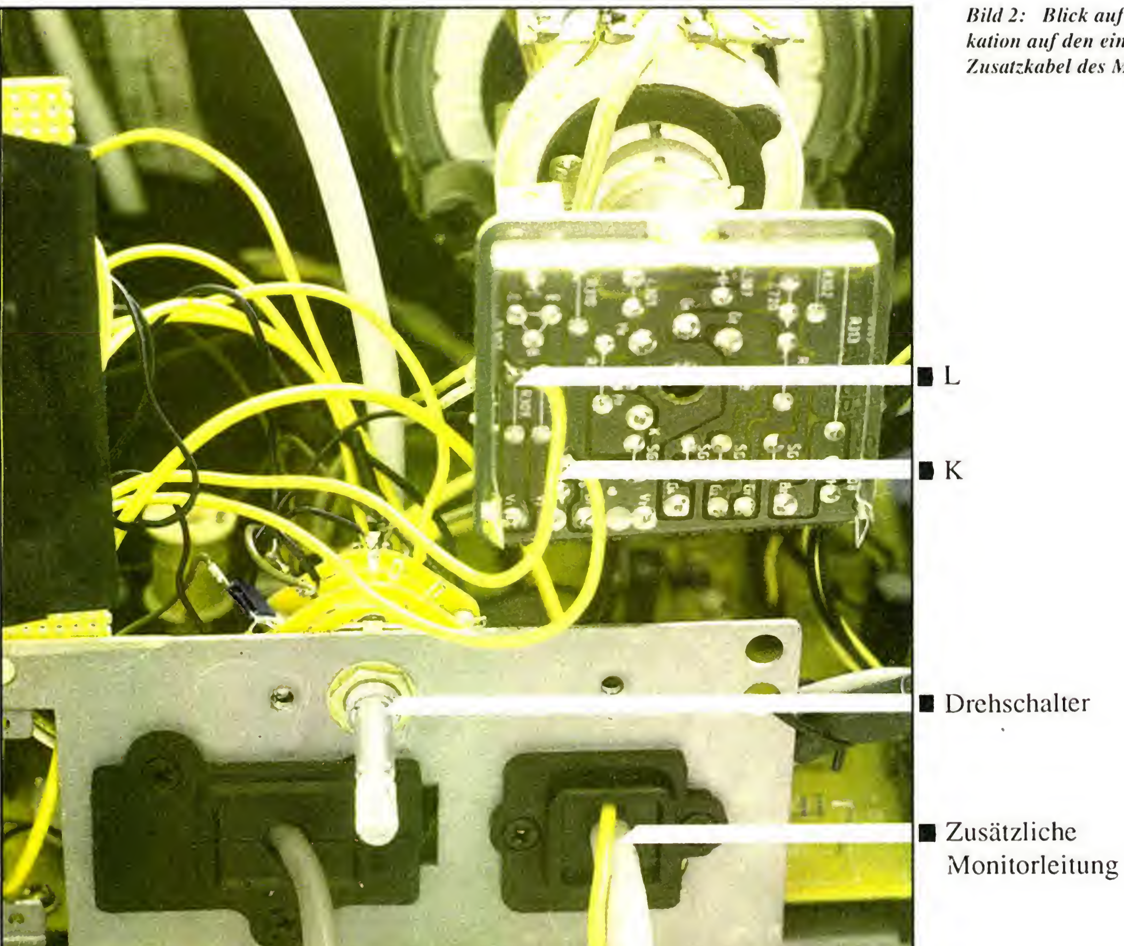


Bild 2: Blick auf den Röhrenhals nach Modifikation auf den eingebauten Schalter und das Zusatzkabel des Monitorsteckers.

J
I
Q
P
C
⊥
E

Bild 3: Zeigt auf der Lötseite der Grundplatine (von unten) die genaue Lötstelle von C2-4, C7-9, R8-11, D3 für Buchstabe A-C, I, N, O, Q).

Bild 4: Linke Platinenseite nach der Modifikation von der Bestückungsseite aus gesehen.

Bild 5: Rechte Platinenseite nach der Modifikation von der Bestückungsseite aus gesehen.

Das Chassis kann danach wieder festgeschraubt werden. (Massestecker der Bildröhre wieder einstecken!)

7.4 Das Antennenkabel kann an den Schalter angelötet werden. Der Analog-Verstärker wird nun auch vollständig verdrahtet. (Buchstabe G, E und das dünne Video-Kabel mit Schirm)

8. Pause

9. Jetzt ist die Umrüstung beendet. Halt! Überprüfen Sie alle Anschlüsse noch einmal. Sind die Kondensatoren C5 und C6 noch auf Ihrem Platz? Werden die Kondensatoren C7, C8 und C9 auch nicht von irgendwelchen Pins aufgespießt? (Kurzschlußgefahr, denn der Zeilentrafo erzeugt Hochspannung!)

10. Alles OK? Wenn "Ja" kann die Modifikation getestet und abgeglichen werden. (Siehe Kasten)

11. Nur wenn Ihnen der neuere Monitortyp gehört, müssen Sie möglicherweise auch ein 70 Volt Netzteil einbauen!

Das ist aber nur erforderlich, wenn Atari auch einen anderen Zeilentrafo eingebaut hat. Im Farbmodus wird dann die Bildhelligkeit viel zu hoch, weil die +70 Volt Spannung auf 50V abfällt.

Das Netzgerät stützt diese Spannung, und somit ist die Helligkeit wieder auf normalem Niveau.

Der Plus-Pol des Hilfsnetzteiles wird an die Kathode von Diode D711 angeschlossen. Der Minus-Pol wird mit Masse (z.B. Chassis) verbunden. An der Kathode von D711 kann man übrigens auch die Gleichspannung messen und somit beurteilen, ob der Einbau des Netzteils nötig ist.

Den Abgleich oder Hilfe bei eventueller Fehlersuche finden Sie in den nebenstehenden Kästen. Wir hoffen, daß Ihnen der Umbau keine Schwierigkeiten bereitet und wünschen viel Spaß mit Ihrem Multisync (Schreiben Sie uns, falls Sie auf besondere Probleme gestoßen sind oder weitere Tips und Tricks für andere Multisynkler haben).

Ralf Bager

Bauteileliste SM 124 Multisync

Netzteil für neues Modell (ab ca. 6/87) nicht inbegriffen!

Widerstände; 5% Toleranz:

1 Miniaturwiderstand 1/8 Watt oder 1/16 Watt, 330 Ω
1 Miniaturwiderstand 1/8 Watt oder 1/16 Watt, 680 Ω
1 Miniaturwiderstand 1/8 Watt oder 1/16 Watt, 1.8 kΩ
Wichtig: Größere Widerstände passen nicht in den Stecker des Monitors.

1 Kohleschichtwiderstand 1/4 Watt, 82 Ω
1 Kohleschichtwiderstand 1/4 Watt, 100 Ω
1 Kohleschichtwiderstand 1/4 Watt, 390 Ω
1 Kohleschichtwiderstand 1/4 Watt, 470 Ω
1 Kohleschichtwiderstand 1/4 Watt, 10 kΩ
1 Kohleschichtwiderstand 1/4 Watt, 3.3 kΩ
1 Kohleschichtwiderstand 1/4 Watt, 470 kΩ
1 Kohleschichtwiderstand 1/4 Watt, 820 kΩ
1 Kohleschichtwiderstand 1/4 Watt, 3.3 MΩ (oder auch 3 x 1 MΩ)

Halbleiter:

4 Diode 1N 4148
1 Universal Gleichrichter-Diode Imax >= 3A Umax >= 50V
1 Diode RGP 30 G oder ersatzweise Diode RGP 15 G
1 Transistor BC 413 B

Kondensatoren:

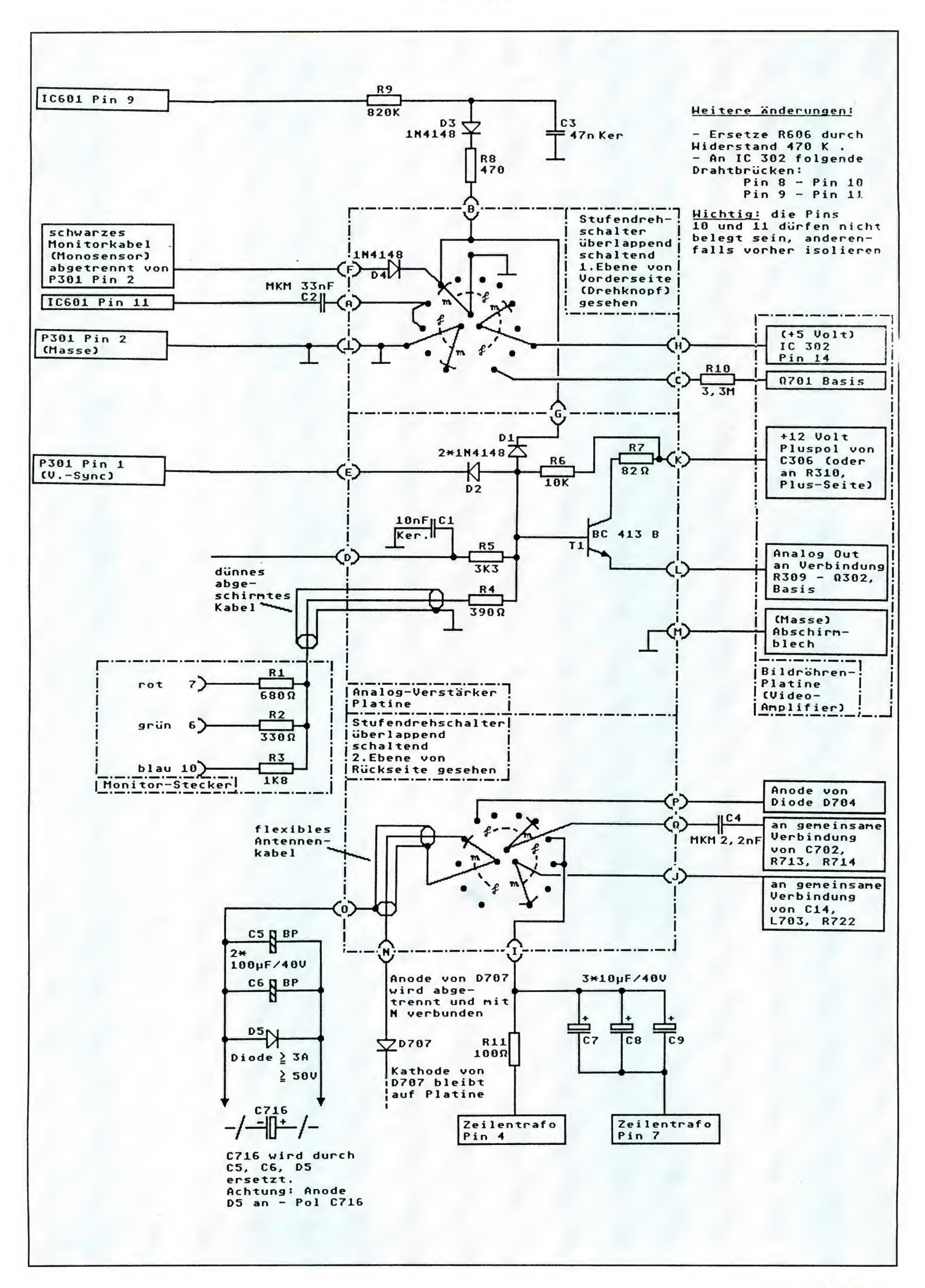
2 Elektrolyt-Kondensatoren ungepolt 100 µF / 40 V (Elkos für Frequenzweiche) (Spoerle Typ EKU 100/40).
Ersatzweise auch 4 x 47 µF / 40 V etc. (werden mit gleichen Polen zusammengeschaltet! Also -C++C- oder +C—C+) Wichtig: Auf gute Qualität achten (kleiner Rers)

3 Elektrolyt-Elkos für Schaltnetzteile 10 µF/ >=40 Volt (Spoerle; EKR 05 10/63 oder 2 x EKR 05 22/40) max. 10mm Duchmesser. Wichtig: Auf gute Qualität achten (kleiner Rers). Schlechte Elkos werden heiß!

1 MKM-Kondensator 2.2 nF
1 MKM-Kondensator 33 nF
1 Ker. Kondensator 47 nF
1 Ker. Kondensator 10 nF

Sonstiges:

1 Drehschalter 6 x 4 Stellungen, überlappend schaltend, Maximalstrom >= 1,5 Ampere. (Salhöfer Elektr.; DS23 6x4 Kontakte).
1 Zierknopf aus PLastik (!) für den Drehschalter
0.5m Antennen-Kabel (50...75 Ω), flexibel.
2 m Abgeschirmtes Kabel, 1-adrig, flexibel, 2.5mm Durchm., Farbe grau. (Am besten ist 1/2 abgeschirmte Zwillingsflachleitung für Stereo-Plattenspieler geeignet).
3 m Bunter Litzenschaltdraht ca. 0.25 mm² (div. Farben)
1 Stk Lochrasterplatine ca. 1.5 x 3 cm oder mehr
20 cm Isolierschlauch 5mm Durchmesser.
10 cm Isolierschlauch 2mm Durchmesser.
20 Kabelbinder grau.



IC601 Pin 9
R9
820K
D3
1N4148
C3
47n Ker
R8
470
Weitere Änderungen!
- Ersetze R606 durch Widerstand 470 K .
- An IC 302 folgende Drahtbrücken:
Pin 8 - Pin 10
Pin 9 - Pin 11
Wichtig: die Pins 10 und 11 dürfen nicht belegt sein, anderenfalls vorher isolieren
Stufendrehschalter überlappend schaltend 1.Ebene von Vorderseite (Drehknopf) gesehen
schwarzes Monitorkabel (Monosensor) abgetrennt von P301 Pin 2
1N4148
D4
MKM 33nF
C2
IC601 Pin 11
P301 Pin 2 (Masse)
(+5 Volt) IC 302 Pin 14
R10
3,3M
Q701 Basis
D1
2*1N4148
R7
82Ω
R6
10K
P301 Pin 1 (V.-Sync)
D2
+12 Volt Pluspol von C306 (oder an R310, Plus-Seite)
10nF C1
Ker.
BC 413 B
T1
R5
3K3
Analog Out an Verbindung R309 - Q302, Basis
dünnes abgeschirmtes Kabel
R4
390Ω
(Masse) Abschirmblech
rot 7
R1
680Ω
grün 6
R2
330Ω
blau 10
R3
1K8
Monitor-Stecker
Analog-Verstärker Platine
Stufendrehschalter überlappend schaltend 2.Ebene von Rückseite gesehen
Bildröhren-Platine (Video-Amplifier)
Anode von Diode D704
C4
MKM 2,2nF
an gemeinsame Verbindung von C702, R713, R714
flexibles Antennenkabel
an gemeinsame Verbindung von C14, L703, R722
C5 BP
2* 100µF/40V
C6 BP
D5
Diode ≧ 3A
≧ 50V
Anode von D707 wird abgetrennt und mit N verbunden
D707
Kathode von D707 bleibt auf Platine
3*10µF/40V
R11
100Ω
C7
C8
C9
C716
Zeilentrafo Pin 4
Zeilentrafo Pin 7
C716 wird durch C5, C6, D5 ersetzt.
Achtung: Anode D5 an - Pol C716

Fehler: Irren ist mänschlich!

1. Falls Ihr Monitor nicht richtig funktioniert - bedenken Sie, Ihr Computer ist nur ein klitzekleiner Teil vom Universum. Stay cool!

2. Versuchen Sie den Fehler zu beheben:

2.1 Wenn der Monitor schon im Monochrom-Modus nicht funktioniert, sind Abgleich und Fehlersuche im Farbmodus wenig angebracht. Der Monitor wurde wahrscheinlich falsch modifiziert.

2.2 Die häufigsten Fehler sind:

- Zu lange Kabel. Die Kabel sollten nicht länger sein, als zur bequemen Montage erforderlich.
- Die Kabel sind farblich nicht zu unterscheiden und deshalb vielleicht vertauscht worden, also ein Verdrahtungsfehler.
- Das Antennenkabel kann abgebrochen sein. Überprüfe außerdem nach kalten Lötstellen.
- Zu dünne Kabel. Es fließen hohe Ströme durch C7-C9, C5,C6,D5 und D707.
- Diode(n) wurden verkehrtherum angeschlossen (auf den Ring achten).
- Der Drehschalter ist verkehrt belegt.
- Minderwertige Bauteile. Die Kondensatoren C5-C9, sowie die Diode RGP 30 G sollten nicht durch irgendwelche Billigware ersetzt werden. C5-C9 darf sich im Betrieb nicht erwärmen. Für die Diode sollte keine Netzdiode genommen werden, da es bei ihr auf schnelle Schaltzeiten ankommt.

2.3 - Wenn das Bild für Monochrom fehlt: Messen Sie mit einem Multimeter das Signal an der Bildröhrenplatine. Es sollte ca. 1,5 V DC sein (abhänig vom Bildinhalt); prüfe die Helligkeitsregler, sowie die Verdrahtung um IC302.

2.4 - Wenn das Bild für den Farbbetrieb fehlt: Messen Sie mit einem Multimeter das Signal an Emitter von T1 (BC 413B). Es sollte ca 1,3 Volt sein (abhängig vom Bildinhalt); untersuche den Analog-Verstärker.

3. Fehler im Farbmodus:

3.1 - Wenn die Gleichspannungen +70V und -70V im Farbmodus nicht stimmen, kommt es u.U. zu falscher Bildhelligkeit. Die Video-Endstufe kann dann übersteuern. Evtl. muß ein Netzteil für diese Spannungen eingebaut werden. Wenn die Bildhelligkeit zu hoch ist, das Bild aber steht, so ist das +70V-Netzteil erforderlich.

3.2 - Andere Fehler mit der Bildhelligkeit sind evtl. auf den Analogverstärker und sein Umfeld zurückzuführen.

3.3 - Bei vertikalen Synchronisationsfehlern sollte der Abgleich überprüft werden. Dann die Bauteile um IC 601 prüfen. Prüfen Sie auch die Verdrahtung im 13-poligen Stecker auf Kurzschlüsse.

3.4 - Überprüfen Sie bei horizontalen Synchronisationsfehlern die Verdrahtung zum R10 und C4.

4. - Wenn die Synchronisation in beiden Richtungen fehlschlägt ist u.U. etwas am Monochromsensor falsch. Beim Umschalten muß jedesmal das Diskettenlaufwerk kurz in Betrieb sein - wie bei einem normalen Reset.

Wie funktioniert's?

Im Farbmodus wird die Vertikalfrequenz (Bildwiederholfrequenz) von 71 Hz auf 50 Hz und die Horizontalfrequenz (Zeilenablenkfrequenz) von 35,7 kHz auf 15,6 kHz verringert.
Durch die Frequenzabsenkung erhöhen sich aber die Ströme in den beiden Ablenkspulen der Bildröhre drastisch. Das Bild würde weit über das normale Bildformat vergrößert. Da Ihr Monitor aber ohnehin nach wenigen Sekunden den Geist aufgeben würde, ist dieses Problem nebensächlich.

Wie sieht also die Gegenmaßnahme aus? Ganz einfach:
Die Ströme werden vor der Umschaltung in den Farbmodus nach und nach abgesenkt. Das erklärt auch die zwei Zwischenstellungen unseres Monitorschalters. Sie bereiten den Monitor gewissermaßen auf seine neue Betriebsform vor.
Der zweite Unterschied zum monochromen Betrieb besteht in der Darstellbarkeit von Grauschattierungen. Da der Monitor von Hause aus nur einen digitalen Eingang für das Monochrom-Signal besitzt, d.h. nur Schwarz oder Weiß darstellen kann, muß ein Analogverstärker zum Mischen der Farbsignale eingefügt werden. Deshalb sind auch die 3 Widerstände im Monitorstecker und der zusätzliche Transistor erforderlich.
Wie sieht nun die Umschaltung im einzelnen aus?
IC 601 ist für die Vertikalablenkung zuständig. Hier sind nur zwei Eingriffe zur Umschaltung erforderlich. Aber der Frequenzunterschied ist ja auch nicht so gewaltig.
Die Horizontalablenkung ist wesentlich komlizierter. Hierfür sind die Transistoren Q 701 bis Q 708, der Zeilentrafo und andere Spulen zuständig. Hier fließt auch ein relativ hoher Strom. Insgesamt müssen durch den Umschalter vier grundlegende Dinge angepaßt werden.
Das TTL Inverter-IC SN 7406 (IC 302) bildet zusammen mit Q 301 und Q 304 den digitalen Videoverstärker nebst Helligkeitsregelung. Sein Ausgangssignal wird mit dem des Analogverstärkers gemischt. Da immer nur einer der beiden Verstärker ein Bildsignal vom Computer erhält, ist eine direkte Umschaltung hier nicht erforderlich.
In der 1. Schalterstellung "mono" bleibt zunächst alles beim alten.
Bereits In der 2. Schalterstellung wird der Monochrom-Sensor (Pin 4 des Monitorsteckers) geöffnet, dadurch schaltet der Computer auf Farbbetrieb. Gleichzeitig wird an IC 601 mittels R9, D3, C3 und R8 die vertikale Bildhöhe reduziert. Die Bildbreite wird durch abtrennen der Diode D707 auf das erforderliche Maß verringert.

geht weiter...



STARKE SOFTWARE

ST ARCHIVAR
DIE ÜBERZEUGENDE DATEIVERWALTUNG

- Unterstützt die Produktion wissenschaftlicher Texte
- Verwaltet und bearbeitet Zitate u. Literaturangaben und Ausgabe als Text
- Komfortable Nutzung für Video- u. Adressendateien
- Datenverwaltung mit Dateien von bis zu 19 Datenfeldern (38 in Vorbereitung und 200 Datensätzen
- Dynamisch erweiterbar
- Veränderung der Dateimasken
- Speicherung der Daten als DIF oder SDF Datei, damit Übernahme in Datenbank möglich
- Beliebige Programme nachladbar u. auszuführen ohne ARCHIVAR zu verlassen (z. B. 1st Word)
- mit allen Textprogrammen kompatibel, die ASCII einlesen
- Ausdruck von Karten (z. B. Bibliothekskarten)
- Listen-Ausdruck auf Knopfdruck
- Etiketten-Ausdruck
- Einfache Druckerprogrammierung
- Einfach bedienbar mit der Maus

ST-ARCHIVAR enthält zahlreiche Hilfsprogramme

- Bibliotheks-Suchfunktion (ganze Disketten durchsuchen lassen)
- Uhr-Einstellung
- Info-Datei über alle ARCHIVAR-Dateien erstellen
- Druckersteuer-Programm
- Kopierprogramm
- Sortierprogramm
- Erzeugt RAM-DISK G bis 1750 KB
- Automatisches Kopieren in die RAM DISK G beim Start

und das alles für **DM 89,-**

ST PRINT
4 NÜTZLICHE PROGRAMME IN EINEM

- **RESETFESTE RAMDISK**
- Größe einstellbar von 32 – 4000 KB
- kann auf Laufwerk C bis P gelegt werden
- arbeitet problemlos mit einer Harddisk
- sehr schnell, da in Maschinensprache programmiert
- **DRUCKERSPOOLER,**
- Größe einstellbar von 2 – 510 KB
- Arbeitet mit TOS- u. GEM-Programmen
- Hohe Geschwindigkeit beim Ausdruck
- **DRUCKERVOREINSTELLUNG**
- mit der Maus, Knopfdruck statt Handbuch
- viele Einstellmöglichkeiten
- Einstellung des Druckers vom Desktop u. aus jedem GEM-Programm (VIP Prof., Wordplus, Tempus etc.) möglich.
- **HARDCOPYROUTINE**
- nutzt die Fähigkeiten von 9-, 18- u. 24-Nadeldruckern
- versch. Auflösungen, Schnelldruck bis Qualitätsdruck
- Umsetzung der Farben in Grauwerte
- gespoolte Hardcopy etc.

DM 59,-

ST PLOT
KURVENDISKUSSIONS- U. FUNKTIONSPLOTTER-PRG.

- ist ein Programm, das mehr kann, als einfach nur Funktionen zeichnen. Es bestimmt die richtigen Ableitungen u. damit werden Null-, Extrem- und Wendestellen berechnet, es stellt den Definitionsbereich u. die Periode fest und vieles andere mehr.
- zeichnet die Schaubilder von bis zu 3 Funktionen u. deren erste u. zweite Ableitung gleichzeitig auf einen Bildschirm. Danach können Ausschnitte der Funktionen vergrößert und verkleinert werden.
- ist komplett GEM- und mausgesteuert. Die Bedienung ist somit sehr einfach. Sie brauchen nur die Funktion einzugeben, alle anderen Berechnungen erledigt für Sie das Programm.
- läuft sowohl in mittlerer wie auch in hoher Auflösung.
- ist nicht nur für Professoren, Studenten u. Schüler interessant. Es kann überall dort angewendet werden, wo man sich mit Funktionen beschäftigt.
- Umfangreiche Eingabemöglichkeiten, z. B. Verwendung sämtl. Funktionen eines wiss. Rechners. Defin. von 10 versch. Zahlenwerten als Konstanten, e u. Pi verwendbar.
- Komfortable Zeichnung, z. B. autom. Zeichnung u. Beschriftung der Achsen und deren Skalierung, Fehlerroutinen, damit keine unnötigen Asymptoten gezeichnet werden, beliebig viele Funktionen nacheinander in ein Bild einzeichenbar ermöglicht Funktionenvergleich u. Ablesung der Schnittpunkte etc.
- Beste Berechnung der Funktionswerte, z. B. Hohe Rechengeschwindigkeit, hohe Rechengenauigkeit.

DM 59,-

ST AKTIE
AKTIENVERWALTUNG

- Aktien- u. Depotverwaltung u. Bilanzierung
- CHART-Analyse incl. Datenbank mit Kurven mit über 50 versch. Aktien beliebig aktualisierbar u. erweiterbar
- Sämtl. großen deutschen Standardwerte (55 St.) bis Okt. 87 fortgeführt
- Kto.-Führung für alle Orders u. andere Buchungen
- Kompakt, übersichtlich, Menü-orientiert, einfache Bedienung durch Maus, u. lfd. Programmhilfen, Graph. Darstellung standardgemäß, generiert selbständig

DM 69,-

ST DIGITAL
LOGIKSIMULATOR FÜR DEN ATARI ST

Ein Programm zum Erstellen, Testen und Analysieren von Logikschaltungen für Ausbildung und Hobby-Elektronik.

- Komfortable GEM-Umgebung
- Bauteile lassen sich per Maus plazieren u. verdrahten
- Umfassendes Bauteile-Set (Grundgatter, Ein-/Ausgabe-Bausteine, Flip-Flops, etc.)
- Definition zusätzlicher Bauteile durch Makrotechnik
- Makros können in Libraries gespeichert werden
- Interaktive Simulation mit Darstellung der Leitungszustände (d. h. Betätigung von 'Schaltern' mit der Maus und sofortige Reaktion der Schaltung)
- Erzeugen von Impulsdiagrammen
- Hardcopy-Funktion
- Hardwarevoraussetzung: ST/Monochrom-Monitor

DM 89,-

SKYPLOT plus
EIN ASTRONOMIEPROGRAMM FÜR STERNENFREUNDE

- zwei Sterndatensätze mit 612 bzw. 15 383 Sternen
- 1053 Nebel, Sternhaufen u. Galaxien
- acht Planeten, Sonne, Mond u. drei Kometen vordefiniert
- Eingabe eigener Bahnelemente möglich
- Darstellung des Himmels als rechteckige Karte
- maßstäbliche Darstellung von Nebeln u. Planeten als Scheiben, letztere als Phasen – etc. dadurch Darstellung von Finsternissen, Durchgängen, Bedeckungen etc.
- automat. Suche nach Konjunktionen
- Bewegungsbahnen von Planeten mit max. 32 000 Punkten, speicherabh.
- einstellbarer Datumsbereich 1.1.1583 – 31.12.3000
- Gradnetz auf Wunsch einblendbar
- für Hardcopy Bild invertierbar
- Darstellung dreidimensional oder von Planeten anderer Sterne aus
- Stellarstatistik; Sichtbarkeitsdiagramme
- Sternkarten können mit allen Objekten auf Plotter ausgegeben werden
- Simulation eines Films im Speicher, dadurch Bewegungsabläufe sichtbar
- Handbuchfunktion verweist bei jedem Menüpunkt auf Handbuchseite
- und vieles andere mehr
- Hardware-Voraussetzung: ATARI ST mit mind. 1 MByte.

DM 198,-
Update DM 80,-

ST-LEARN
DER VOKABELTRAINER FÜR DEN ST

programmiert v. D. Owerfeldt, Gewinner der GOLDENEN DISKETTE '87 für das beste Lernprogramm.

- Läuft vollständig unter GEM
- „Intelligente" Auswertung der Benutzereingaben
- Fehlerhäufigkeit einer Vokabel wird berücksichtigt
- Berücksichtigung mehrerer Bedeutungen eines Wortes
- Vielfältige Möglichkeiten des Lernens und der Abfrage
- Integriertes Lernspiel „HANGMAN"
- Spezielle Auswertung für unregelmäßige Verben (bei Eingabe von „to go" werden die anderen 2 Formen nachgefragt)
- Bei offensichtlicher Ähnlichkeit der Wörter wird wahlweise ein 2. Versuch zugelassen
- Trotz Einordnung der Vokabeln nach Lektionen oder Wissensgebieten ständig schneller Zugriff auf **alle** Vokabeln (nur durch Größe des Speichermediums begrenzt)
- Voller europäischer Zeichensatz (Zugriff durch die Maus unter GEM)
- Auch für Farbmonitor in mittlerer Auflösung
- Wörterbuchfunktion durchsucht alle Files eines Speichermediums nach einer Übersetzung ab
- Verbessertes Eingabeformular

DM 59,-

ST STRUKTUR-PAINTER
DAS SUPER-PROGRAMM zum kreativen Erstellen von Graphiken aller Art – ohne Malkenntnisse

- Einfache Bedienung, ansprechende Menüs
- Automatischer Malmodus, Muster in Millionen Variationen erstellt
- 2 Füllroutinen, 2 Blockspeicher
- WYSIWYG
- Ausschneiden u. Einkopieren von beliebigen Formen
- Freihand, Verschiedene Pinsel, Füllmuster als Pinsel, Malen mit Bildausschnitten
- Füllmuster definieren ohne Editor; eine ganz neue Methode
- Füllmuster-Bibliotheken; über 200 Füllmuster gleichzeitig verfügbar
- Vergrößern, Verkleinern; 3 Endprodukte zur Auswahl
- Spiegeln, Drehen (dreht in 2 Richt. gleichzeit.)
- Stauchen
- Rahmen (1): Erzeugen von Rahmen aus allen Graphiken
  Rahmen (2): Automatische Rahmenberechnung
- Schatten: Automatische Schattenberechnung (3-D-Effekt)
- Teilmuster; Font's (23 Größen, 21 Arten, 4 Verknüpfungen
- Lineal: Einblendbare Einteilung
- Radierer u. Sprühdose beliebig einstellbar
- Druckertreiber für alle Epson- u. IBM-kompat. Drucker sowie Laserdrucker implementiert
- Schnelle Lupe mit: Punkt, Rechteck, Linie, Invertieren, Löschen
- Invertieren auf Knopfdruck während gezeichnet wird
- Voll Mausgesteuert
- Keine Kompatibilitätsprobl. mit Bildern anderer Programme
- STruktur-Painter Bilder auch mit Signum 2 verarbeitbar.
- und viele weitere Möglichkeiten
- Hardwarevoraussetzung: ST mit mind. 512 KB frei

Ein Spitzenprogramm **DM 89,-**

· alle Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON an Heim-Verlag Heidelberger Landstraße 194 6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ______

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______

Straße, Hausnr. ______ PLZ, Ort ______

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

An den Anschlüssen 2-5 vom Zeilentrafo sinkt dadurch die Betriebsgleichspannung auf ca. 40%; Diode D5 ist nun stromführend und hält die Gleichspannung an Pin 4 von Z.-Trafo auf 11,4 Volt.
In der 3. Schalterstellung wird IC 601 mittels C2 auf ca. 55 Hz umgeschaltet. Das ermöglicht den 60 Hz Betrieb. IC 601 paßt sich automatisch an die Vertikalfrequenz vom Computer (50 oder 60 Hz) an. Parallel zu C713 wird C7-C9 geschaltet. Damit ist die horizontale Bildverzerrung einigermaßen angepaßt.
In der 4. Schalterstellung wird der Oszillator für die Horizontalfrequenz von 35,7 kHz auf 15,6 kHz umgeschaltet (C4). Damit dieser auch mit dem Computer synchronisiert, tritt gleichzeitig R10 in Aktion.

Beim Umschalten von Farbe auf Monochrom läuft alles in umgekehrter Richtung ab. Der Schalter darf also bei eingeschaltetem Gerät betätigt werden. Theoretisch muß aber jede Stufe für 0,05 Sekunden geschaltet sein.

Für den eigentlichen Abgleich muß auf jeden Fall ein s.g. Abgleichbesteck - das sind Schraubendreher verschiedenster Größe aus Bernstein (teuer) oder Plastik - verwendet werden. Mit Metallschraubendrehern funktioniert ein Abgleich nicht (das Metall verändert die Induktivität einer Spule)! Zum Abgleich geht man folgendermaßen vor:

1. Zunächst wird die bisherige Position der Einstellregler mit Filzstift markiert. Das ist wichtig, falls nach einem Abgleichfehler der ursprüngliche Zustand wiederhergestellt werden soll. Dazu gehört aber ein Protokoll, da einige Regler mehr als eine Umdrehung ermöglichen.
2. Bildfangregler einstellen (VR 601).
3. Im Monochrom-Modus das Bildformat 21.7 cm x 13.5 cm mittels L 702 und VR 602 einstellen.
4. Evtl. L 703 ca. 2mm im Urzeigersinn verdrehen. (Für einen Kompromiß zwischen Farb- und S/W-Betrieb)
5. Bildverzerrung vertikal (VR 603) kontrollieren.
6. In Farbmodus umschalten und Bildverzerrung bzw. Höhen / Breiten-Verhältnis kontrollieren. *Die horizontale Bildverzerrung ist nicht 100-%ig. Bei genauem Hinsehen erkennt man, daß gleiche Icons ganz links etwas breiter sind - es stört jedoch kaum.*

Bezeichnung	Zweck/Hinweise
VR 602 (V.Size)	Stellt Gesamthöhe des Bildes ein. *Muß leicht nachgestellt werden.*
L 702 (H.Size)	Diese Spule stellt die Gesamtbreite des Bildes ein. *(Muß nur verstellt werden, wenn das Bild leicht vergrößert werden soll).* *Bildhöhe und Breite sollten im richtigen Verhältnis zueinander stehen, damit aus einem Kreis nicht eine Ellipse wird, bzw. aus einem Quadrat kein Rechteck.*
VR 603 (V.Line)	Behebt die vertikale Bildverzerrung. *Nur dann verstellen, wenn z.B. Objekte am Bildrand (oben/unten) zu hoch, in der Bildmitte aber normal dargestellt werden.*
L 703 (H.Line)	Behebt die horizontale Bildverzerrung. *Hier gibt es ein Problem mit der Farbauflösung:* *Die Bildverzerrung ist nicht vollständig zu beheben. Der verbleibende Fehler stört jedoch nicht sehr. Außerdem sollte man evtl. einen Kompromiß zwischen Monochrom- und Farbbild suchen. Man muß dazu die Einstellschraube ca 2mm auf einer imaginären Skala im Urzeigersinn verdrehen.*
VR 601 (V.Hold)	Bildfangregler für stillstehendes Bild. *Wenn verstellt ergeben sich Paternoster-Effekte.* *Stellen Sie den Regler so ein, daß der Monitor 50Hz, 60Hz und 71Hz (monochrom) fängt.*
VR 701 (H.Center)	Hiermit wird das Bild in die Mitte des Schirms gebracht. Im Monchrom-Modus einstellen.
VR 703 (Focus) Vorsicht: Hochspannung 600 V	Für die Bildschärfe. Wird so eingestellt, daß man jedes Pixel genau erkennt. *Die meisten Bildröhren haben am äußeren Bildrand unscharfe Bereiche. Je größer man das Bild also einstellt, umso weniger Details sieht man.* *Ein Kompromiß ist bei etwa 21.7 cm Breite und 13.5 cm Höhe in monochrom erreicht. Man stellt dann die Bildschärfe so, daß sich der Rand schärfer abbildet, ohne daß in Bildmitte ein auffallend unscharfer Bereich entsteht.*
VR 702 (Sub Bright)	Grundeinstellung für die Helligkeit. (Sollte wegen der Bildröhren-Lebensdauer nicht verändert weden).

Da alle Abgleicharbeiten am geöffneten Monitor durchgeführt werden müssen, denken Sie in erster Linie an Ihre eigene Sicherheit. Verwenden Sie keinerlei Metallschraubendreher. Das Berühren der Bauteile, Platinen und Bildröhre kann zu einem tödlichen Stromschlag führen! Verwenden Sie einen Trenntrafo.

ENDE

Karl Vogel
GEM Programmierung im Modula-2
Romanshorn 1988
AXIS Verlag
120 Seiten
DM 25,-

Wie der Titel schon sagt, ist das Buch für Modula-2-Programmierer gedacht, die GEM-Applikationen schreiben wollen. So orientiert sich die Darstellung des AES und VDI an den Modulen von TDI-Modula-2. Vorausgesetzt wird die Beherrschung der Sprache und des TDI-Systems.
Im ersten Kapitel wird zunächst eine übersichtliche Einführung in das GEM-Konzept gegeben. Da für richtige GEM-Programme unerläßlich, beschreibt Vogel dann die Benutzung des Resource-Construction-Sets im zweiten Abschnitt, um daraufhin mit den Alertboxen (die er seltsamerweise "Sofortboxen" nennt) in die eigentlichen GEM-Routinen einzusteigen. Es schließen sich die Dialogboxen mit besonderer Berücksichtigung der TEDinfos an.
Im vierten Kapitel findet sich der nächste Bestandteil des AES wieder - die Menüs. Von der Erstellung eines Menübaumes kommt der Autor zur eigentlichen Menüprogrammierung und schneidet hier auch schon kurz den Event-Manager an.
Als dritter AES-Bestandteil belegen die Fenster das nächste Kapitel. Dabei kommen alle wichtigen Prozeduren von der Abfrage des Bildschirmhandles bis zum eigentlichen Öffnen des Fensters und den Fenster-Events vor. Bei den Routinen unterläuft dem Autor allerdings ein kleiner Fehler, denn ein Programm braucht nach einem WindowOpen keineswegs die Fensterfläche zu löschen, dafür erzeugt GEM eine Redraw-Mitteilung.
Gut gelungen sind allerdings die nützlichen Hilfsroutinen für Fenster, die, in ein kleines Modul gepackt, praktisch brauchbar sind. Ebenfalls nützlich ist die Beschreibung der Fensterroutinen aus dem TDI-Toolkit, da es hierfür kein richtiges Handbuch gibt (s. ST-Computer 11/86).
Abschließend wird noch kurz auf einige der elementaren VDI-Routinen eingegangen, die bei der GEM-Programmierung notwendig sind.
Der Autor wird dem selbstgestellten Anspruch gerecht, ein kleines Lehrbuch zu schreiben, das das Wesentliche der GEM-Programmierung erläutert. Der Text ist didaktisch gut aufgebaut und angenehm zu lesen. Die Beispielprogramme sind passend ausgewählt und leiten weiter zu eigenen Experimenten.
Allerdings hätte ich mir an einigen Stellen mehr Informationen gewünscht. 120 Seiten reichen eben nur für eine Einführung; allerdings verweist Vogel auch in der Einleitung auf weiterführende Literatur. Im Zeitalter der halbautomatischen Registererstellung hätte das Buch auch einen Index verdient und sollte nicht abrupt mit einer Abbildung der VDI-Füllmuster enden.
"GEM-Programmierung in Modula-2" ist dennoch eine gelungene Einführung für den Einsteiger, der sich mit dem TDI-Modula-Compiler gerüstet auf den Atari ST stürzen will und ein auf sein System zugeschnittenes Werk sucht.

Robert Tolksdorf

Ivan Bratko
PROLOG
Programmierung
für künstliche Intelligenz
Addison Wesley Verlag

Dieses Buch ist die deutsche Übersetzung der englischen Originalausgabe Prolog Programming for Artificial Intelligence, die 1986 im gleichen Verlag erschienen ist. Und dieses gehörte zu meiner Lieblingslektüre im Bereich der KI, weil es einerseits Grundlagen der Sprache PROLOG vermittelt, andererseits aber auch recht komplexe Anwendungen bereithält. Man kann dem Übersetzer bestätigen, daß dem Buch nichts von der Faszination des Originals verlorengegangen ist.
Der erste Teil des Buches beschäftigt sich mit der Arbeitsweise des Prolog-Interpreters. Wohltuend registriert der Neuling, daß die Ausführungen hier und im ganzen Buch nicht akademisch trocken dargeboten werden, sondern reichlich mit Beispielen garniert sind. Zur Selbstkontrolle finden sich nach jedem Kapitel Übungsaufgaben, deren Lösung der Leser am Ende des Buches in einem Anhang findet. Damit ist das Buch auch Autodidakten zum Selbststudium zu empfehlen. In sieben Kapiteln mit über 200 Seiten erhält der Leser einen Überblick über Prolog, erfährt etwas über Instanzierung und Matching (der Autor vermeidet den aus der Logik stammenden Begriff Unifikation), lernt mit Listen umzugehen, übt den Gebrauch von Strukturen wird mit der Kontrolle des Backtracking vertraut gemacht, der Anfängern immer wieder Schwierigkeiten bereitet. Kapitel über Ein-Ausgabe, vordefinierte Prozeduren und Programmierstil und -technik runden den ersten Teil ab. Schon in diesem Teil finden sich so interessante Programme wie das Acht-Damen-Problem oder die Simulation eines nichtdeterministischen Automaten. So richtig interessant wird es dann im zweiten Teil.
Der zweite Teil des Buches trägt die Überschrift Prolog in der künstlichen Intelligenz und deutet damit an, daß wir uns dem Kern des Buches nähern. In 8 Kapiteln wird der Leser mit Problemen vertraut gemacht, die in konventionellen Programmiersprachen nicht oder nur schwer zu lösen sind. Die Themenbandbreite reicht von einfachen Operationen auf Datenstrukturen über heuristische Suchverfahren und Expertensysteme bis hin zum musterorientierten Programmieren. Dieses Kapitel hat mich am meisten fasziniert, wird hier doch ein Programm diskutiert, das in der Lage ist, logische Beweise selbständig auszuführen.
Der Inhalt jeden Kapitels wird am Schluß nochmals stichwortartig zusammengefaßt. Eine wahre Fundgrube ist auch die dort gegebene Bibliographie, die dem Leser die Möglichkeit gibt, sich mit der Originalliteratur auseinanderzusetzen.
Der Autor des Buches verwendet die DEC10-Syntax. Dieser Dialekt ist der einzige für den ATARI ST verfügbare PROLOG-Dialekt. Damit dürfte es den Besitzern des TOY-Prologs aus der PD und des SALIX-Prologs sowie des MPROLOGs problemlos möglich sein, den Ausführungen des Autors zu folgen.
Dieses Buch ist wegen der klaren Gliederung, den vielen instruktiven Beispielen und der interessanten Thematik jedem wärmstens zu empfehlen, der sich mit dem Gedanken trägt, in den Problemkreis der KI oder PROLOG einzutreten. Der zweite Teil des Buches dürfte allerdings auch erfahrenen PROLOG-Anwendern zumindest Interesse abverlangen.

Dr. K.Sarnow

ANALYSE DER DRUCKERAUSGABE

Exceptions in Modula-2

Moderne Drucker bieten eine Option an, die es ermöglicht, empfangene Zeichen und Steuercodes in einer Hexdump-ähnlichen Weise auszugeben. Wer diese Zusatzfunktion bei seinem Drucker vergeblich sucht, kann auf ein Accessory zurückgreifen, das diese Aufgabe übernimmt.

Das Accessory belegt einen Eintrag namens "Drucker Hexdump" im Desk-Menü. Öffnet man die Meldebox, kann man jederzeit die Sonderfunktion an- oder ausschalten. Beim Ausschalten wird jedesmal ein CR und LF an den Drucker geschickt, um den Puffer zu entleeren. Dies ist notwendig, da die Steuerzeichen nicht ausgeführt werden und der Drucker eine unvollständige Zeile erst ausgibt, wenn er ein CR bekommt. Das Programm soll gleichzeitig ein Beispiel dafür sein, wie man die Umlenkung der Exception-Vektoren von einer Hochsprache ausprogrammieren kann.

Das Hauptprogramm

Das Hauptprogramm trägt sich in die Accessory-Liste ein und wartet zunächst auf eine Meldung vom GEM. Falls die Accessory Open-Meldung eintrifft, wird nach Setzen des richtigen Defaults die Box eröffnet. Wählt der Anwender den An-Knopf zum ersten Mal, wird mit Hilfe der BIOS-Funktion GetException der alte Trap 13-Vektor eingelesen und gesichert. Dabei ist zu beachten, daß dies der 45. Systemvektor ist. Danach wird mit SetException der neue Vektor eingetragen. Die Hilfsvariable active verhindert ein Überschreiben des gesicherten Vektors für den Fall, daß der An- Knopf ein zweites Mal betätigt wird. Schaltet man die Option wieder aus, wird der alte Exception-Vektor restauriert,so daß der Prozessor bei einem BIOS-Aufruf wieder direkt ins Betriebssystem springt.

Die Prozedur newexception

Ist die neue Funktion aktiviert, landet der Prozessor bei jedem Trap 13 am Beginn der Prozedur newexception. An dieser Stelle schafft sich der Compiler Platz auf dem Stack, um seine lokalen Variablen unterzubringen. Dabei bedient er sich des Maschinenbefehls link x,a6, wobei für x eine negative Zahl eingesetzt wird um die sich der Stackpointer (SP) erniedrigt. Vorher wird jedoch der Inhalt von Register a6 auf den Stack gebracht und der SP in a6 geschrieben. Als nächstes sichert man am besten alle restlichen Register auch auf dem Stack. Das Register a4 wird noch einmal extra gesichert, da später für den Rücksprung noch ein Register benötigt wird. An die Parameter des BIOS-Aufrufs kommt man, indem man den in a6 abgelegten SP plus 4 betrachtet. Dieser Wert wird dem Zeiger auf das Record zugewiesen, so daß elegant auf die Parameter zugegriffen werden kann. Zunächst muß nun überprüft werden, ob es sich überhaupt um einen BConOut-Aufruf handelt, der ausgeführt wird, sobald irgendein Programm eine Ausgabe tätigt. Dazu muß man die Funktionsnummer auf 3 überprüfen. Ist nun auch noch der Drucker durch den Device 0 angesprochen, verzweigt das Programm in die neue Ausgabefunktion. Die ConvertHex-Prozedur wandelt den Zeichencode in das hexadezimale Format um. Zur besseren Darstellung wird bei einstelligen Hex-Zahlen noch etwas korrigiert. Die Ausgabe kann nun nicht über Standardprozeduren erfolgen, da diese wieder rekursiv in newexception landen würden. Deshalb wird die Prozedur out benötigt. Der Zähler counter wird bei jedem Durchlauf um 1 erhöht und bewirkt bei Erreichen einer Marke das Ende einer Zeile auf dem Drucker durch Ausgabe eines CR- und LF-Codes. Nach der Ausgabe muß man für den Rücksprung sorgen. Zunächst werden alle Register bis auf a6 vom Stack geholt. Anschließend besorgt die unlk a6-Operation die gleiche Stackkonfiguration wie am Anfang der Prozedur, und rte-Befehl. erledigt den Rücksprung. Bleibt noch, einen Sprung ins Betriebssystem vorzusehen, falls kein BConOut auf den Drucker erfolgen sollte. Dazu werden auch zuerst die Register wiederhergestellt, bevor die unlk-Operation durchgeführt wird. Nun wird das Register a4 benötigt, welches die Adresse oldexception aufnimmt, um sie auf den Stack zu bringen. Danach ist nach Wiederherstellung von a4 nur noch ein rts-Befehl nötig, um den Sprung auszuführen. Dabei ist zu beachten, daß a4save eine globale Variable ist, damit SETREG(12,a4save) in einen einzigen move.l-Befehl übersetzt werden kann und keine Register mehr benutzt.

geht weiter...

Die Prozedur out

Die Prozedur out gibt die Zeichen aus, indem ein BIOS-Aufruf simuliert wird. Es werden dabei die Parameter entsprechend auf dem Stack bereitgestellt, bevor ein Sprung in die alte Betriebssystemprozedur ausgeführt wird.

Als erstes muß man Platz auf dem Stack für die BIOS-Parameter schaffen. Dazu ermittelt man mit der Systemfunktion TSIZE die Länge des Records und erniedrigt den Stack um diesen Wert. Gleichzeitig wird der Zeiger bios geladen. Als nächstes werden die Parameter in die Liste eingetragen. Dabei muß man auch die Rücksprungadresse ermitteln, da am Ende der Betriebssystemprozedur ein rte-Befehl steht. Das bedeutet, daß Statusregister und Programmzähler (PC) vom Stack geladen werden, so daß man nicht mit einem jsr-Befehl einspringen kann. Zunächst wird durch einen bsr2(PC)-Befehl ein Unterprogrammaufruf simuliert, der freundlicherweise ein Speichern des PCs auf dem Stack bewirkt. Dabei wird aber lediglich der nop-Befehl übersprungen. Als nächstes wird der PC vom Stack nach a3 gebracht und nach Addition von 28 als Rücksprungadresse abgelegt. Die 28 entsteht aus dem Abstand zwischen dem bsr-Befehl und der Stelle, wo der Rücksprung landen soll. Man kann dies an den beigefügten Längenangaben der Befehle im Programm nachvollziehen. Die letzten 3 Maschinenbefehle dienen wieder zum Einspringen in die Betriebssystemprozedur. Hat diese ihre Arbeit getan, muß der Stack korrigiert werden. Der SP steht zu diesem Zeitpunkt vor biosnr, denn Status und PC wurden bereits vom Stack geholt. Deshalb wird nur len-6 zum SP addiert.

Hoffentlich ist das Prinzip klar geworden. Man kann den Rumpf der Prozeduren, insbesondere die maschinensprachlichen Teile, für ähnliche Aufgaben wie z.B. Drucker-Spooler, Drucker-Anpassung an Sonderzeichen direkt übernehmen. Es ist durch einfaches Ändern des Records möglich, auch andere BIOS-, XBIOS- oder GEMDOS-Funktionen abzufangen und zu verändern. Es sollte jedoch vermieden werden, innerhalb der neuen Exception-Prozedur Betriebssystemfunktionen aufzurufen, da dies in der Regel zu Abstürzen führt.

Michael Schaffner

```
43 C  1E .  77 w  75 u  72 r  64 d  65 e  6E n  1E .  62 b  65 e  72 r  65 e
69 i  74 t  73 s  1E .  76 v  6F o  6D m  1E .  53 S  74 t  61 a  63 c  6B k
1E .  67 g  65 e  68 h  6F o  6C l  74 t  2E .  1E .  1C .  44 D  65 e  73 s
68 h  61 a  6C l  62 b  1E .  77 w  69 i  72 r  64 d  1E .   D .   A .  6E n
75 u  72 r  1E .  6C l  65 e  6E n  2D -  36 6  1E .  7A z  75 u  6D m  1E .
53 S  50 P  1E .  61 a  64 d  64 d  69 i  65 e  72 r  74 t  2E .   D .   A .
 D .   A .  48 H  6F o  66 f  66 f  65 e  6E n  74 t  6C l  69 i  63 c  68 h
1E .  69 i  73 s  74 t  1E .  64 d  61 a  73 s  1E .  50 P  72 r  69 i  6E n
7A z  69 i  70 p  1E .  6E n  75 u  6E n  1E .  6B k  6C l  61 a  72 r  1E .
67 g  65 e  77 w  6F o  72 r  64 d  65 e  6E n  2E .  1E .  4D M  61 a  6E n
1E .  6B k  61 a  6E n  6E n  1E .  64 d  65 e  6E n  1E .  52 R  75 u  6D m
70 p  66 f  1E .   D .   A .  64 d  65 e  72 r  1E .  50 P  72 r  6F o  7A z
65 e  64 d  75 u  72 r  65 e  6E n  2C ,  1E .  1C .  69 i  6E n  73 s  62 b
65 e  73 s  6F o  6E n  64 d  65 e  72 r  65 e  1E .  64 d  69 i  65 e  1E .
6D m  61 a  73 s  63 c  68 h  69 i  6E n  65 e  6E n  73 s  70 p  72 r  61 a
63 c  68 h  6C l  69 i  63 c  68 h  65 e  6E n  1E .  54 T  65 e  69 i  6C l
65 e  1E .  66 f  81 .  72 r  1E .   D .   A .  84 .  68 h  6E n  6C l  69 i
63 c  68 h  65 e  1E .  1C .  41 A  75 u  66 f  67 g  61 a  62 b  65 e  6E n
2C ,  1E .  1C .  7A z  2E .  42 B  1E .  1C .  44 D  72 r  75 u  63 c  6B k
65 e  72 r  1E .  53 S  70 p  6F o  6F o  6C l  65 e  72 r  2C ,  1E .  1C .
```

Bild: So wird der Hexdump auf dem Drucker ausgegeben

```
 1:  IMPLEMENTATION MODULE DumpPrt;
 2:
 3:       (*  Dies ist ein Desk Accessory und muß *)
 4:       (*  deshalb mit GEMACCX.LNK an Stelle von
              GEMX.LNK gelinkt werden !  *)
 5:
 6:  (*$A+,$T-,$S-*)   (* Code optimieren,
                          keine Bereichsüberprüfung *)
 7:
 8:  FROM GEMAESbase          IMPORT AccessoryOpen;
 9:  FROM AESApplications     IMPORT ApplInitialise;
10:  FROM AESEvents           IMPORT EventMessage;
11:  FROM AESForms            IMPORT FormAlert;
12:  FROM AESMenus            IMPORT MenuRegister;
13:  FROM SYSTEM          IMPORT TSIZE,ADDRESS,ADR,CODE,
                                 SETREG,REGISTER,LONGWORD;
14:  FROM BIOS            IMPORT GetException,SetException,
                                 BConOut,PRT;
15:  FROM M2Conversions IMPORT ConvertHex;
16:  FROM Terminal        IMPORT WriteString,WriteLn;
17:  FROM ASCII           IMPORT CharIsPrintable;
18:  (* So sieht der Stack aus, wenn man nach einem
        BConOut-Aufruf i.d. Exception-Prozedur ankommt. *)
19:  TYPE stackconfig  =  RECORD  status : CARDINAL;
20:                               pc     : ADDRESS;
21:                               biosnr : CARDINAL;
22:                               device : CARDINAL;
23:                               dummy  : CHAR;
24:                               char   : CHAR
25:                           END;
26:        biosstack      =   POINTER TO stackconfig;
27:
28:  VAR   handle,counter,
29:        choice,default,
30:        applID,menuID          : INTEGER;
31:        a4save,oldexception    : ADDRESS;
32:        poldexc                : PROC;
33:        AlertText              : ARRAY[0..79]OF CHAR;
34:        MessageBuffer          : ARRAY[0..7] OF CARDINAL;
35:        active                 : BOOLEAN;
```

```
36:
37:    PROCEDURE out(c:CHAR); (* Ausgeben eines Zeichens
                                 durch Aufruf der alten
                                 BIOS-Prozedur *)
38:    VAR bios:biosstack; len:ADDRESS;
39:    BEGIN
40:       len := TSIZE(stackconfig); (*Länge d.Parameter-
                                       liste feststellen *)
41:       bios:= REGISTER(15)-len; (* Vom akt.Stack-
                                      Pointer abziehen *)
42:       SETREG(15,bios);          (* Neuen SP wieder
                                       abspeichern      *)
43:                                (* und Parameter ein-
                                       tragen:           *)
44:       bios^.status := 2000H; (* Supervisor Mode nach
                                    Rückkehr *)
45:       bios^.biosnr := 3;      (* BConOut *)
46:       bios^.device := 0;      (* Printer *)
47:       bios^.char   := c;      (* Das Zeichen       *)
48:       bios^.dummy  := 0C;     (* Vorsichtmaßnahme *)
49:                               (* Rücksprungadresse
                                     berechnen  *)
50:       CODE(06102H);      (* bsr    2(pc)  pc -> sp  *)
51:       CODE(04E71H);      (* nop                 +2  *)
52:       CODE(0265FH);      (* move.l (sp)+,a3     +2  *)
53:       bios^.pc:=REGISTER(11)+28; (*            +14  *)
54:       SETREG(12,oldexception);
                             (* move.l oldexc,a4    +6  *)
55:       CODE(02F0CH);      (* move.l a4,-(sp)     +2  *)
56:       CODE(04E75H);      (* rts                 +2  *)
57:       (* Hier müssen wir nach der Ausgabe ankommen *)
58:       SETREG(15,REGISTER(15)+len-6); (* Stack-
                                           Korrektur  *)
59:    END out;
60:
61:    PROCEDURE newexception;
62:    VAR  bios:biosstack; hex:ARRAY[1..2] OF CHAR;
63:    BEGIN                  (* link     a6          *)
64:       CODE(048E7H,0FFFCH); (* movem.l d0-a5,-(a7) *)
65:       a4save:=REGISTER(12); (* move.l  a4,a4save   *)
```

```
 66:        bios:=REGISTER(14)+4; (* move.l  a6+4,bios   *)
 67:        IF (bios^.biosnr=3) AND (bios^.device=0) THEN
                                (* BConOut Printer? *)
 68:           ConvertHex(ORD(bios^.char),2,hex);
 69:           IF hex[2]='H' THEN hex[2]:=hex[1];
                  hex[1]:=' ' END;
 70:           out(hex[1]);
 71:           out(hex[2]);
 72:           out(' ');
 73:           IF CharIsPrintable(bios^.char)
                  THEN out(bios^.char)
 74:              ELSE out('.') END;
 75:           out(' ');
 76:           IF counter=12   (* 13 Hex-Blöcke pro
                                  Zeile *)
 77:              THEN out(15C); out(12C); counter:=0
 78:              ELSE out(' '); INC(counter) END;
 79:           CODE(04CDFH,03FFFH);
                              (* movem.l (a7)+,d0-a5    *)
 80:           CODE(04E5EH);  (* unlk     a6             *)
 81:           CODE(04E73H);  (* rte                     *)
 82:        ELSE
 83:           CODE(04CDFH,03FFFH);
                              (* movem.l (a7)+,d0-a5    *)
 84:           SETREG(12,oldexception);
                              (* move.l  oldexception,a4 *)
 85:           CODE  (04E5EH); (* unlk    a6             *)
 86:           CODE  (02F0CH); (* move.l  a4,-(a7)       *)
 87:           SETREG(12,a4save);
                              (* move.l  a4save,a4      *)
 88:           CODE  (04E75H); (* rts                    *)
 89:        END
 90:     END newexception;
 91:
 92:     PROCEDURE ConvLongs(A:LONGWORD;VAR B:LONGWORD);
 93:     BEGIN
 94:        B:=A;
 95:     END ConvLongs;
 96:
 97:  BEGIN
 98:     applID:=ApplInitialise();
 99:     active:=FALSE;
100:     menuID:=MenuRegister(applID,'  Drucker Hex Dump');
101:     AlertText:='[2][ Drucker Dumping|     von
                          |Michael Schaffner][An|Aus]';
102:     LOOP
103:        EventMessage(ADR(MessageBuffer));
104:        IF MessageBuffer[0]=AccessoryOpen THEN
105:           IF active THEN default:=2
106:                     ELSE default:=1 END;
107:           choice:=FormAlert(default,AlertText);
108:           IF (choice=2) AND active THEN
                (* Aus *)
109:              ConvLongs(oldexception,poldexc);
110:              SetException(45,poldexc); (* alten Vektor
                                            restaurieren  *)
111:              active := FALSE;
112:              BConOut(PRT,15C);BConOut(PRT,12C) END;
                                  (* CR und LF ausgeben *)
113:           IF (choice=1) AND NOT active THEN   (* An   *)
114:              counter:=0;
115:              oldexception:=GetException(45); (* alten
                                  Trap 13 Vektor sichern *)
116:              SetException(45,newexception);  (* neuen
                                  Trap 13 Vektor setzen  *)
117:              active := TRUE END;
118:        END;
119:     END;
120:  END DumpPrt.
```

ENDE

RELAX

Hallo lieber Spielefan !

Der Atari ST ist ein hervorragender 16-Bit-Computer mit guten Möglickeiten zur Erstellung von Grafik und Sound. Aufgrund dieser Voraussetzungen eignet er sich bestens zur Programmierung technisch brillianter Spielesoftware. "ST-Computer" hat die Computerspiele bisher stark vernachlässigt. Ich möchte ab dieser Ausgabe den Lesern so viele ST-Spiele wie möglich auf den dazu zur Verfügung gestellten Seiten vorstellen. Dazu wurde der zweigeteilte Spieleteil eingeführt. Im ersten Teil lesen Sie ausführliche Spielereviews, im zweiten Teil gibt es Kurzvorstellungen und News. Ich möchte in der nächsten Ausgabe auch eine Hitparade veranstalten und Wettbewerbe arrangieren. Unter jeder Nachricht und jedem Review ist ein Info angegeben. Die aufgeführten Vertriebsfirmen geben Ihnen Auskunft, wo Sie die getesteten Produkte in Ihrer Nähe beziehen können. Viel Spaß beim Lesen wünscht Ihnen:

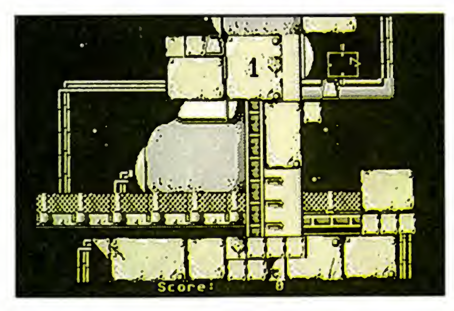
Carsten Borgmeier



NEWS

PREDATOR

Arnold Schwarzenegger ist der Protagonist von ACTIVISIONS neustem Actionspiel für den ST. Das Spiel heißt PREDATOR. Die Handlung orientiert sich an der des gleichnamigen brutalen Kinofilms. Sie laufen bewaffnet mit einer Panzerfaust



durch den Dschungel und müssen einige wichtige Persönlichkeiten aus den Klauen von Entführern befreien. Im Dschungel wimmelt es nur so von feindlichen Soldaten, die auf Arnold mit dem Gewehr anlegen. Man läuft von links nach rechts über den Bildschirm und ballert,was das Zeug hält. Dabei muß man Munition sammeln und sich nach durchschlagskräftigeren Waffen umsehen. Predator ist ein sehr brutales Spiel, bei dem auf Menschen geschossen werden muß. Das ist sicher nicht jedermanns Sache. Vielleicht fällt das Spiel auch der Indizierung zum Opfer. Verständlich wär's.

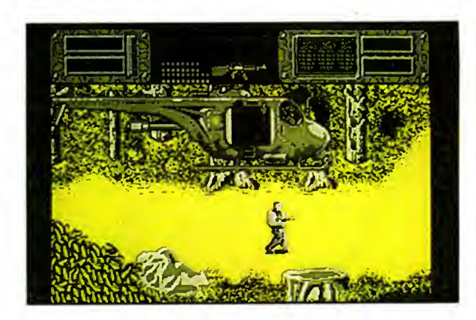
Ariolasoft, Leisuresoft

TERRAMEX

GRAND SLAM ENTERTAINMENTS bietet TERRAMEX, ein Spiel mit lustigen Grafiken und einem gelungenen Sound für den St an. Des Spielers Aufgabe ist sehr schwierig: Ein Meteroit stürzt auf Mutter Erde zu. Professor Eyestrain, der einzige Mensch, der diese Katastrophe verhindern könnte, lebt zurückgezogen in seinem Labor, irgendwo auf der weiten Welt. Mit dem Joystick steuern Sie Ihren Helden durch die Abenteuerwelt, um den Professor zu finden. Dabei müssen verschiedene Gegenstände eingesammelt werden, um das Spiel zu lösen. Mit einem Staubsauger kann man beispielsweise in den Himmel fliegen und dort wichtige Gegenstände finden. Terramex beschert aufregende Computerspielstunden.

Ariolasoft

BOB MORAN

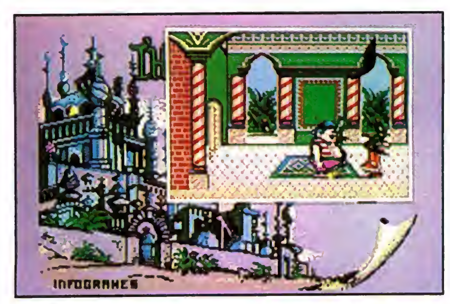


BOB MORAN SCIENCE FICTION von INFOGAMES ist ein Actionspiel, das sehr an ein indiziertes Spiel erinnert und ebenfalls von Infogames stammt. Mit der Maus bewegt man ein Fadenkreuz über eine futuristische Landschaft. Wenn man einen Feind erspät, muß man ihn mit dem Fadenkreuz anvisieren und in die ewigen Jagdgründe schicken. Das Spiel ist in einer aufwendigen Hintergrundstory verpackt. Nette Grafiken und ein passabler Sound rechtfertigen einen Kauf.

Profisoft, Ariolasoft

ISNOGUD

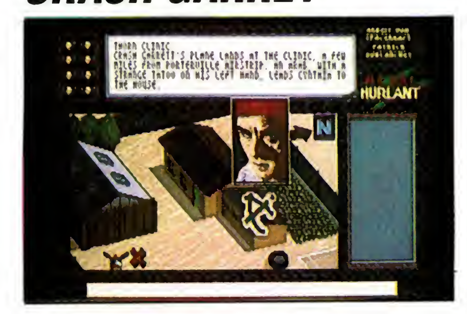
In einem kleinen Fenster auf dem Bildschirm findet die Handlung zu Isnogud von Infogames statt. Der Ort des Geschehens ist Bagdad. Der Spieler soll in Gestalt eines Taugenichts mit Namen Isnogud den Kalifen von seinem Thron vertreiben und die Re-



gierungsgeschäfte selbst in die Hand nehmen. Ein komplett mit Joystick gesteuertes Arcadeadventure der Spitzenklasse.

Ariolasoft, Profisoft

CRASH GARRET



In die Zeit des Nationalsozialismus versetzt Sie ein Grafikadventure aus der Welt der Piloten und Spione. Intelligente Texte, hervorragende Grafiken und ein guter Sound machen CRASH GARRETT von INFOGAMES spielenswert.

Info: Leisuresoft

ROCKFORD

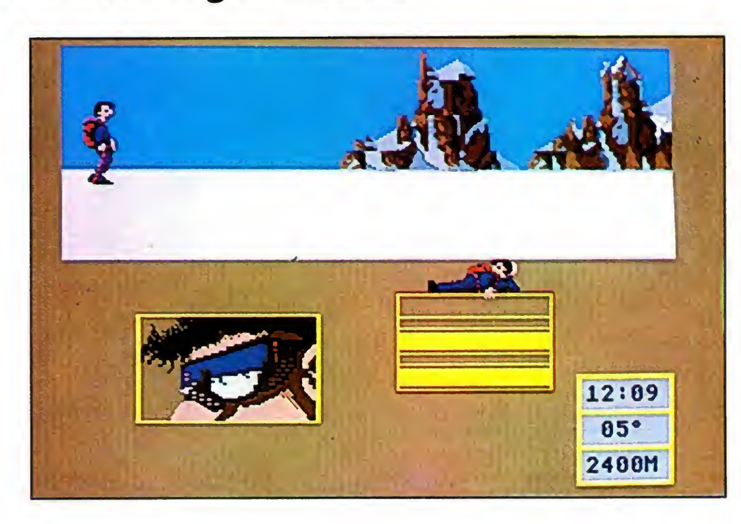


Kennen Sie den Diamantensammler Rockford aus den Homecomputerspielen der Boulder Dash Serie, die für den C64 zu haben ist. Der quirlige Diamantensammler bewegt sich durch neue Höhlenlabyrinthe und kämpft gegen andere Widersacher. ROCKFORD von MELBOURNE HOUSE ist eine 1:1 Umsetzung des gleichnamigen Automaten. Rockford ist die beste Boulder Dash Variante, die je geschrieben wurde. Ein Klassespiel ! Get it !

Leisuresoft

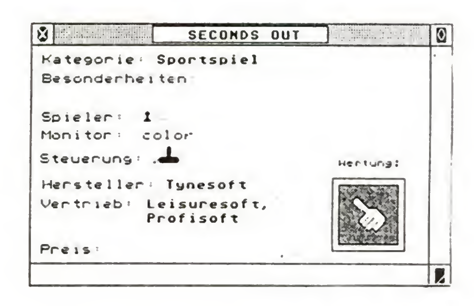
CHAMONIX CHALLENGE

Kletterpartie in luftigen Höhen



Schmackhafte Softwarekost kommt nicht nur aus England, Amerika oder Deutschland. Immer mehr hervorragende Spiele werden aus unserem Nachbarland Frankreich importiert. Viele Toptitel Made in France stammen von Infogames, dem Softwarehaus, dem wir "Sidewalk" und "Reisende im Wind" zu verdanken haben. Infogames veröffentlichte vor kurzem die Atari ST Version von Chamonix Challenge, einer waschechten Bergsteigersimulation, in dem Strategie und vor allem eine geschickte Hand am Joystick verlangt werden, wenn es darum geht, in den Bergen von Chamonix die Gipfel zu bezwingen. Das Spiel ist sehr komplex, so daß es sehr erfreulich ist, daß sowohl Programmtexte als auch Anleitung in Deutsch verfasst worden sind. Nach dem Laden der Disk erscheint ein Menü, in dem man eine von sechs Aufstiegsrouten auswählt. Ist die Frage der Route geklärt, geht es weiter mit dem Rucksackpacken. Es erscheint eine Liste mit vielen Dingen, die man auf den Berg mitnehmen kann. Durch häufiges Spielen lernt man, welche Gegenstände man unbedingt benötigt und auf welche man getrost verzichten kann. Daß Schlafsack, Seile, Schneebrille und Haken für eine Bergbesteigung offensichtlich wichtig sind, ist klar, also einpacken ! Es ist nicht nur wichtig, die richtigen Kletterutensilien einzupacken, sondern auch warme Klamotten mitzunehmen, um auf dem Berg nicht zu erfrieren. Der Spieler hat das Schicksal des Bergsteigers voll in seiner Hand. Er muß dafür Sorge tragen, daß der mutige Kletterer bei schwierigen Steilwänden im richtigen Tonus einen Fuß vor den anderen setzt und sich mit einem Seil absichert, um nicht in die Tiefe zu stürzen.

Die Besteigung des Berges ist sehr schwierig, weil man sich erst im Erklimmen von Steilwänden üben muß. Chamonix Challenge macht viel Spaß, wenn man Zeit für eine langwierige Kletterpartie mitbringt. Die Grafik haut zwar keinen Bären vom Barhocker, ist jedoch für eine Bergsteigersimulation angemessen. Schnee ist nun mal durchweg weiß. Wenn Ihnen Chamonix Challenge genauso gefällt wie mir, wird das Spiel ein Gipfelstürmer.



SECONDS OUT

Kategorie: Sportspiel
Besonderheiten:

Spieler: 1
Monitor: color
Steuerung:

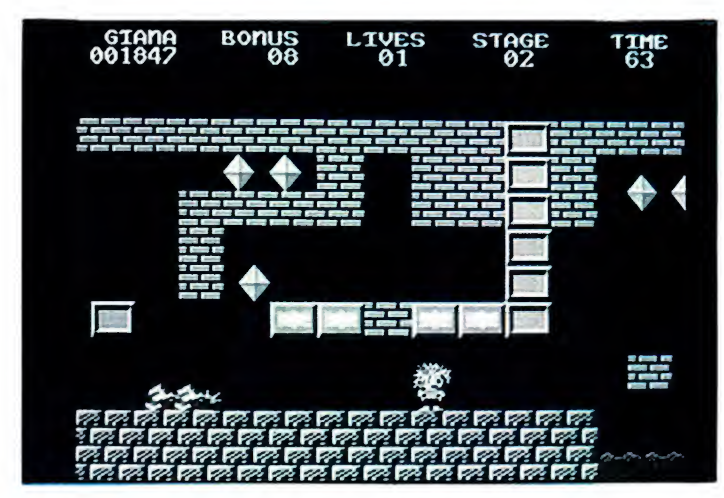
Hersteller: Tynesoft
Vertrieb: Leisuresoft, Profisoft

Preis:

Wertung:

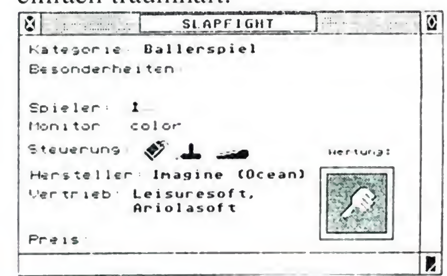
GIANA SISTERS

Vorsicht Suchtgefahr!



Ich habe selten ein Spiel gesehen, das den Spieler so lange an den Bildschirm fesselt wie GREAT GIANA SISTERS von RAINBOW ARTS. Nachdem das Actionspiel schon Spielefans auf dem C64 und auf dem Amiga verzauberte, gibt es endlich auch eine Atari ST Version. Die originelle Handlung wurde selbstverständlich von den Vorversionen übernommen: Giana, ein kleines Mädchen aus Genua, liegt friedlich in ihrem Bettchen und träumt, als sie plötzlich etwas Sonderbares bemerkt. Ihr Traum ist Realität geworden. Giana befindet sich in einer wunderbaren Welt, in der die Gesetze der Schwerkraft beinahe aufgehoben sind. Sie kann riesige Sprünge machen und schwebt dabei für einige Zeit in der Luft. Einfach herrlich ! Doch nach einigen Stunden will sie wieder nach Hause in ihr Bettchen. Nur, wie kann sie die Traumwelt verlassen ? Giana öffnet die Verpackung des Spiels, greift zur Anleitung und da steht, daß sie ohne einen Riesendiamanten mit Zauberkraft nicht mehr in die reale Welt zurückkehren kann. Also macht sich das kleine Mädchen auf die Suche. Als Spieler und ehrlicher Käufer des Abenteuers sitzt man am Joystick und ist ihr dabei behilflich. Die Traumwelt besteht aus 30 verschiedenen Abschnitten, die Giana durchreisen muß. Im 26.Level befindet sich der Zauberdiamant, den Giana zur Rückkehr braucht. Auf das Mädel in Genua wartet eine gefährliche Reise. Monsterhafte Gestalten wie Riesenskorpione, Spinnen und Tellermonster wollen nichts Gutes. Tiefe Schluchten, Feuer und siedend heißes Wasser sind ebenfalls eine große Gefahr. Giana läuft behutsam durch die Traumwelt, springt auf Plattformen und sammelt Diamanten ein. Für 100 Diamanten bekommt sie ein Extraleben. Damit Giana ihr schwer erarbeitetes Extraleben nicht so schnell wieder verliert, sind in der Traumwelt viele Extrawaffen versteckt. Wenn Giana unter bestimmte Felsen springt, fliegt eine solche heraus, die sie dann nur noch einsammeln muß. Neben den Bonusgegenständen gibt es aber auch noch andere Überraschungen. Versteckte Schatzhöhlen und Warp Zones, die Giana in einen anderen Level katapultieren, machen das Spiel lange interessant. Technisch ist The Great Giana Sisters hervorragend. Die Grafik ist farbenprächtig und detailreich. Der Sound hört sich im Vergleich zur Amigaversion an einigen Stellen sogar noch besser an.

Die Titelmelodie klingt auf einigen Monitoren recht dumpf. Bei solchen Monitoren muß man den Sound voll aufdrehen, um alles verstehen zu können. Da es beim Atari ST schwierig ist, ein horizontales Scrolling zu programmieren, scrollt der Bildschirm nicht wie bei der Amigaversion. Die einzelnen Bilder werden nur rasant umgeschaltet. Hat man sich an diese Methode erst einmal gewöhnt, steht dem Spielspaß nichts mehr im Wege. Rainbow Arts hätte für dieses Spiel einen Software Oskar verdient, auch wenn man die Spielidee vom Automatenspiel Super Mario Brothers abgekupfert hat. Das Plagiat ist einfach traumhaft.



PINK PANTHER

macht den ST unsicher !



Paulchen Panther dürfte nicht nur jüngeren Lesern ein Begriff sein. Das rosarote Tierchen trieb vor einigen Jahren in einer Zeichentrickserie sein Unwesen. Als Comicheld mußte Paulchen sich ebenfalls schon behaupten. Wo der Panther auch auftaucht, das Herz des Publikums erobert er im Sturm.





DAS SUPER-PROGRAMM ZUM KREATIVEN ERSTELLEN VON GRAPHIKEN ALLER ART – OHNE MALKENNTNISSE

- Läuft auf allen ST mit min. 512 K Ram und dem Mega ST mit Monocrome Monitor (unterstützt Blitter wenn vorhanden)
- **Ein Superding!**
  Ein Programm zum kreativen Erstellen von Graphiken aller Art, ohne Malkenntnisse. Einfache Bedienung. Ansprechende Menüs. Alle Bildoperationen lassen sich rückgängig machen. Automatischer Malmodus, der Muster in Millionen Variationen erstellt.
- **2 Füllroutinen**
- **2 Blockspeicher**
- **Absolut flimmerfreie Block und Bildverarbeitung**
  Die eingestellte Verknüpfung ist schon beim Zeichnen und bewegen von Bildteilen zu sehen.
- **WYSIWYG!**
  Blocks lassen sich in jeder Richtung über den Bildrand hinausschieben. **Ausschneiden und Einkopieren** von beliebigen Formen!
- **Freihand**
  Verschiedene Pinsel. Füllmuster als Pinsel. Malen mit Bildausschnitten. Spiegeln beim zeichnen um 1 oder 2 Achsen
- **Füllmuster definieren ohne Editor !!**
  eine ganz neue einfache aber wirkungsvolle Methode
- **Füllmuster-Bibliotheken**
  über 200 Füllmuster gleichzeitig verfügbar
- **Vergrössern**
- **Verkleinern**, 3 Endprodukte zur Auswahl
- **Spiegeln**
- **Drehen:** dreht in 2 Richtungen gleichzeitig
- **Stauchen**
- **Rahmen (1):** Erzeugen von Rahmen aus allen Graphiken
- **Rahmen (2):** automatische Rahmenberechnung
- **Schatten:** automatische Schattenberechnung (3D Effekt)
- **Teilmuster:** aus Bildteilen (und Teilchen) können weitere Muster, auch Füllmuster erstellt werden.
- **Font's:** 23 Grössen, 21 Arten, 4 Verknüpfungen, weitere Verarbeitung durch Schatten, Rahmen usw. möglich, Schreibrichtung und Farbe kann während des Schreibens geändert werden.
- **Lineal:** Einblendbare Einteilung
- **Radierer und Sprühdose** in jeder Grösse einstellbar. Zwei Sprühdosen Modi.
- **Druckertreiber** für alle Epson und IBM-Kompatiblen Drukker sowie Laserdrucker implementiert.
- **Schnelle Lupe** mit: Punkt; Rechteck; Linie; Invertieren; Löschen
- **Weitere Optionen:** z. B. Bewegen, Kopieren, Kopieren nach..., Rechteck, Kreis, Linie, 3D-Rechteck, Preistafel...
- **Invertieren** auf Knopfdruck während gezeichnet wird
- **Voll Mausgesteuert**
- Spricht Laufwerke „A" bis „F" an
- Keine Kompatibilitäts-Probleme mit Bildern anderer Programme, auch Colorbilder werden selbstständig umgewandelt.
- **STruktur Painter** Bilder können **Signum 2** verarbeitet werden!
- Für Designer, Werbestudios, Druckgewerbe, Graphiker, aber auch für den nicht professionellen Anwender z. B. zum Erstellen einer Schülerzeitung, Handzettel, Grußkarten usw.

Ein Spitzenprogramm **DM 89,–**

ST-ARCHIVAR
DIE ÜBERZEUGENDE DATEIVERWALTUNG

Das Programm, das die Produktion wissenschaftlicher Texte erleichtert, Zitate und Literaturangaben bearbeitet und als Text ausgibt, oder auch zur Behandlung von Adressen- und Videodateien

- Enthält eine Dateiverwaltung mit Dateien von bis zu 19 Datenfeldern und 200 Datensätzen. Die Dateigrößen sind bis zur Speichergrenze dynamisch erweiterbar
- Bildung von Selektionsdateien. Nachträgliche Veränderung der Dateimasken
- Kompatibel mit allen Textprogrammen, die ASCII einlesen
- Einfache Bedienung mit der Maus
- Flexible Ausgabe als Text, Etikette, Karte, Liste. Spezielle Druckermasken über Textprogramme erstellbar
- Komfortable Formatierung für die Textausgabe der Daten, die das Format für bibliographische Angaben erfüllen sollen
- Speichert Daten auch als DIF oder SDF Datei, dadurch in Datenbanken einlesbar
- Schnittstelle zu Datenbanken, dadurch die Ausgabefunktion für Literaturangaben verwendbar
- Arbeitet mit RAM DISKS oder HARD DISKS
- Erlaubt Laden und Ausführen beliebiger Programme ohne ST ARCHIVAR zu verlassen (Diskettenformatierung!)
- Bietet an, eigene Druckersteuerungen zu editieren, die für die Druckerausgabe des Desktops verwandt werden können.
- Beschleunigt die Arbeit. Beim Start des ST kann eine Ramdisk erstellt und Disketten komplett mit Daten- oder Programmdateien eingelesen werden
- Läuft auf allen ATARI's – auch auf MEGA-ATARI

Da auch der nicht textorientierte Anwender in dem Programmpaket praktische Hilfsprogramme findet, ist ST ARCHIVAR jedem Anwender zu empfehlen.

DM 89,–

ST-MATH
DAS PROGRAMM FÜR SYMBOLISCHE ALGEBRA UND ANALYSIS

- rechnet ohne Rundungsfehler
- verarbeitet symbolische Ausdrücke wie 2x + 3y = 5z
- löst Gleichungen nach beliebigen Variablen auf
- beherrscht Grenzwerte, Differential- und Integralrechnung
- ideal für Matritzenrechnung
- unglaublich schnell, da **vollständig** in Assembler geschrieben
- nicht nur ein Mathematik-Programm, sondern gleichzeitig eine komplette KI-Sprache, die leicht erlernbar ist
- die Kombination von Mathematik-System und Programmiersprache ermöglicht auch Anfängern komplexe Mathe-Programme mühelos zu schreiben
- einfache und komfortable Bedienung
- für Schüler, die sich Fehlrechnungen ersparen wollen
- für Lehrer, die mit ST-MATH Klausuren mühelos und schnell korrigieren wollen
- für Studenten, die lange Umformungen und Rechnungen zeitsparend durchführen wollen
- für Ingenieure und sonst. Anwender, die oft komplexe nichtnumerische Probleme lösen müssen
- für Jeden, der eine preiswerte, aber dennoch vollwertige Sprache für künstliche Intelligenz sucht, die leicht zu erlernen ist
- Hardware: ATARI ST mit 512 KB u. ROM-TOS oder 1 MB (dann auch RAM-TOS möglich), eins. Disk.-Laufw.
- arbeitet mit Farb- und SW-Monitoren
- Lieferung mit umfangreichem deutschen Handbuch, das die Mathematikfunktionen detailliert erläutert und zugleich eine ausführliche Einführung in die Programmierung von ST-MATH enthält.

DM 98,–



DAS DEPOTVERWALTUNGS- UND CHART-ANALYSEPROGRAMM FÜR AKTIONÄRE MIT HOHEN ANSPRÜCHEN

Wollen Sie auch GELD an der Börse verdienen?

- Verwaltung von bis zu **100** Aktien mit bis je **300** Kursen.
- Verwaltung von ausländischen Aktien mit bis zu **16** verschiedenen Währungen. (mit mathematisch genauer Einstandswährung.)
- Wertpapiere können alphabetisch sortiert werden.
- Mischen von bis zu **20** verschiedenen Depotdateien.
- **5** verschiedene Kurzfristcharts und **2** verschiedene Langfristcharts.
- Auf Mausklick übereinanderlegen von verschiedenen Langfristcharts in weniger als **0,5** Sekunden (und natürlich wieder zurück).
- Charts können mit eigenen Kommentaren versehen werden. (beliebig viele!!!)
- Beliebig **gleitender Durchschnitt** (beliebig viele gleichzeitig).
- Sie können im Chart **zeichnen.** (Erkennen Sie Trendkanäle, Unterstützungen oder Widerstände.)
- **RSI-CHART** auf Mausklick. (RSI = Relative Stärke Index – damit arbeiten die Profis.)
- **DEGAS** kompatibel (Sie können alle Charts auf Diskette abspeichern und mit Graphicprogrammen weiterarbeiten.) Dies erleichtert Ihre Argumentationstechnik wesentlich.
- **DELUXE** unterstützt natürlich auch **Kapitalerhöhungen** und **Dividendenzahlungen.** (mit Operation Blanche!!!), und zeichnet diese natürlich im Langfristchart ein!
- Umfangreiches graphisches **Hilfsmenü** implementiert.
- **2** verschiedene **Kurseingabe**möglichkeiten. (Für jeden Bedarf das Richtige!!!)
- Die wichtigsten Börsenusancen abrufbar.
- Alle Daten im Speicher = keine langwierigen Diskettenoperationen.
- Festplattenkompatibel.
- Programm wurde von **Aktienhändler** geschrieben.
- **Deutsches Handbuch. Up-date-Service.**
- Erforderliche Hardware: **ATARI ST** mit min. **1 MB Ram. Monochrome Monitor SF 354** oder kompatible.

DM 398,–

alle Preise sind unverbindlich empfohlene Verkaufspreise

Erhältlich bei Ihrem Computer-Händler

Programme, die helfen



MAGIC BYTES, das erfolgreiche Softwarehaus aus Gütersloh, hat die Lizenz von Paulchen für ein Computerspiel ersteigert. PINK PANTHER ist ein absoluter Superhit. Eine lustige Handlung, exellente Comicgrafik und ein pfiffiger Sound wissen den Spieler zu unterhalten. Paulchen ist mal wieder völlig blank. Ein Job als Hausdiener käme ihm da sehr gelegen. Tagsüber könnte er Staub wischen und seinem Chef die Pantoffeln herbeitragen und nachts, wenn der reiche Knopf schläft, könnte er das Haus ausräumen. Er bräuchte nur fünf Häuser auszuräumen, dann könnte er sich ein angenehmes Leben im Süden leisten. Doch bevor seine Träume wahr werden, muß er erst zum Arbeitsamt und sich empfehlen lassen. "Für den Job brauchen Sie nur einen Zylinder, der Hausherr legt Wert auf Äußerlichkeiten". Diese Szenen sind natürlich köstlich animiert. Wenn Paule stolzer Besitzer eines Zylinders ist, kann die Arbeit beginnen. Tagsüber wischt er Staub und nachts... Oh nein ! Der Hausherr ist Schlafwandler. Paulchen muß bei seiner "nächtlichen Räumungsaktion" darauf achten, daß der Gute nicht aufwacht. Während Paul alle Wertgegenstände einsackt, muß er gleichzeitig darauf achten, daß der Hausherr nicht gegen Wände läuft oder am Teppichrand stolpert. Zu diesem Zwecke hat Paulchen ein kleines Glöckchen. Immer, wenn Paule bimmelt, läuft der Hausherr in seine Richtung. Außerdem kann er Gegenstände wie kleine Brücken und Gummipanther aus seinen Taschen kramen und aufblasen, die sein Opfer auf den richtigen Weg bringen. Als Spieler sitzt man schweißgebadet am Joystick, sammelt in dem geräumigen Haus, das sich über mehrere Etagen erstreckt, Wertgegenstände ein und sorgt panisch dafür, daß der Hausherr nicht aus seinem Schlafwandlerschlaf aufwacht. PINK PANTHER halte ich für eins der besten Computerspiele, das jemals geschrieben wurde. Grafik, Sound, Idee, man hätte es nicht besser machen können. Da das Spiel auch noch ziemlich kniffelig ist, verspricht Pink Panther lange Zeit beste Computerspielunterhaltung.

Infoadressen:

Ariolasoft GMBH	***Leisuresoft***	***Profisoft***	***Rushware***
Postfach 1350	*Industriestr. 23*	*Sutthauser Straße 50/52*	*Bruchweg 128-132*
4830 Gütersloh 1	*4709 Bergkamen 5*	*4500 Osnabrück*	*4044 Kaarst 2*
Tel: 05241/803871	*Tel: 02389/6071*	*Tel: 0541/53905*	*Tel:02101/6070*

ENDE

10 MEGAHERTZ NACHLESE

Tja, 10 Megahertz per Nachlese, was hat es damit auf sich? Zunächst einmal möchten wir uns herzlich für die Flut an Leserbriefen, Anrufen und sonstigen Mitteilungen bedanken, die durch diesen Artikel ausgelöst wurden. Doch dann müssen wir mit ein bißchen schlechtem Gewissen gestehen, daß wir unsere Leser in den April geschickt haben, wie diejenigen, die das Listing richtig abgetippt haben, an dem "April, April" unschwer erkennen konnten!

Bevor Sie jetzt wüste Morddrohungen, Verwünschungen u.ä. loswerden wollen, möchten wir zumindestens auf einen Punkt hinweisen, der jedem noch so Unerfahrenem mit etwas kriminalistischem Spürsinn hätte auffallen sollen; und zwar handelt es sich dabei um das Datum, das im Listing steht. Wie soll der gute Herr Tolksdorf denn ein Programm am 1.4.88 (1.April) erstellt haben, wenn die April-Ausgabe der ST-Computer schon vorher auf den Markt kam. Außerdem sollte man generell in einer April-Ausgabe eines Computer-Magazins etwas vorsichtig sein; denn auch in der c't, der Kickstart usw. kann man bei unkritischem Lesen leicht in den April geschickt werden. Auf dieser Seite wollen wir Ihnen nun ein paar der Reaktionen vorstellen (Übrigens handelt es sich hierbei um keinen Virus!):

Die armen Leute, die diese Data-Wüste abgetippt haben, um das Allerletzte aus ihrem ST rauszuholen... (Vorschlag für zukünftiges FAST-Register: $FFFA31 - der MFP belegt nur ungerade Adressen !) Interessieren würde es mich schon, was das Programm WIRKLICH macht ...

Th.W.

Speziell möchte ich mich bei Ihnen bzw. bei Ihrem Mitarbeiter Herrn Tolksdorf für das super Programmlisting CLOCK UP.ACC im Beitrag "10 Megahertz per Software" bedanken. Ich habe ausgiebige Tests durchgeführt und festgestellt, daß einige rechenintensive Programme wohl etwas schneller werden, aber die 30%ige Steigerung, wie im Beitrag angegeben, leider nicht erreichen. Bei 1st_Word Version 2.02 wird das Programm eher langsamer. Ich bitte Sie, mir zu berichten, ob das eventuell an der Version meines 1st_Word liegen kann.

N.K.

An R.Tolksdorf:
Betr.: 10 Megahertz!

Spitze! Aber begegne mir ja nicht in einer dunklen Straße, ich garantiere für nichts.

MfG J.-M.K.

Ausgezeichnet, ganz ausgezeichnet! Selten so gelacht...
"10 Megahertz per Software". Ha! Jetzt können mich die ganzen IBM-Fans mit ihren 6, 8, 10, 12 MHz mal! Mein ST wird nun auch mal aufgemotzt. Ich träume von schnellerer Graphik, Turbo-1st_Word und Formel-1-Compilern.
Nachdem ich mich mit ST Pascal+ anstatt BASIC und Diskettenmonitor durch die BASIC-Data-Wüste gequält habe (... 41 50 52 49 4C, tatsächlich, das heißt ja "APRIL"; was'n Zufall ...), verzweifelt die Fehler gesucht (verdammt, warum stimmt die Checksumme immer noch nicht) und mich dann zu einem Testlauf entschlossen habe, dachte ich zuerst an einen Absturz übelster Art. April, April! Das darf doch wohl nicht wahr sein! Doch ein Scherz dieses Kalibers ist wirklich die Stunden in der Datenwüste wert, kommt auch eine ganz andere Erfrischung als erwartet.
Mit herzlichen Hacker-Grüßen

Chr.K.

Diesen rücksichtslosen Artikel hätte ich eher in einem Micky Mouse Heft als in einer Fachzeitschrift erwartet. Da ich mich als Leser ernst genommen fühlte, habe ich mich sofort daran gemacht 1122 Data-Zeilen abzutippen, wofür ich gute 3 Stunden brauchte. Weitere 2 Stunden gingen für die Fehlersuche drauf, das macht 5 Stunden Arbeit, wobei ich einen steifen Hals und viereckige Augen bekam. Stellen Sie sich vor, ich würde Sie für 5 Stunden arbeiten lassen und danach April, April sagen.
In Enttäuschung

Th.K.

Auch per Modem ließen sich einige interessante Dinge zu dem Thema auffangen:

Kennt schon einer den April-Scherz in der DOS International? In der ST war es das Beschleunigungsprogramm auf 10 MHz. War doch offensichtlich.

Sag mal, was ist denn das für ein Clockup-Prg. in der ST-Computer 4/88? Wohl ein April-Scherz? Kannste das nicht mal hier ablegen?

Hi all!
Ich kenne einen, der das richtig abgetippt hat.

Was für ein Gesicht macht er denn inzwischen?

Jetzt hat er sich schon beruhigt, aber damals weinte er leise vor sich hin oder schrie so etwas wie "die mache ich kalt". Er hatte für das Abtippen etwa 2 Stunden gebraucht.

ENDE

SOFTWARE DIE ES IN SICH HAT!

Nun auch in Deutschland!

Schwarz auf Weiß kann jetzt jeder auf seinem ATARI ST unter MS-DOS arbeiten. Doch nicht nur monochrom, nein, auch in Farbe ist nun der Zugriff auf die Welt der PC-Rechner möglich.

NEU EURO VERSION 3.64

Die Software-Emulation **PC ditto** öffnet allen ATARI ST Anwendern das Tor zum gewohnten professionellen Business Standard.

Mit dem **PC ditto** können Sie mühelos mit Lotus 1-2-3 oder Symphony Ihre Kalkulation erstellen, oder Ihre Daten mit DBase III plus verwalten.

Mit dem **PC ditto** haben Sie Zugang zu Turbo Pascal, zum GW BASIC Interpreter, und auch zu Borlands neuestem Kind Turbo Basic.

Mit dem PC ditto läuft auch die Software, die es für den ATARI ST noch gar nicht gibt.

Mit dem **PC ditto** laufen so viele Programme, daß wir sie hier gar nicht alle auflisten können.

Der **PC ditto** unterstützt die ATARI-Festplatte, den Druckerport, sogar den Laserdrucker und alle Schnittstellen, soweit es die Hardware zuläßt.

Machen Sie aus Ihrem ATARI ST den preiswertesten PC-Clone! Bestellen Sie den **PC ditto** zum Superpreis von nur DM 198,-.

Wir laden Sie ein! Steigen Sie mit Ihrem ATARI ST in die Welt des MS-DOS ein.

Hiermit bestelle ich _ PC ditto für 198.-

Anruf genügt!
Tel.: 06196/481811, Mo-Fr 9-13 und 14-17 Uhr. Schriftliche Bestellung nur gegen Vorkasse oder Nachnahme (Versandkosten DM 7,50; bei Nachnahme zuzüglich DM 3,50 Nachnahmegebühr)

AUSLAND NUR GEGEN VORKASSE

Name:________ Vorname________

Straße:________ Ort:________

Unterschrift:________

MS-DOS und GW-BASIC sind Warenzeichen von Microsoft Corp. Lotus 1-2-3+ Symphony sind Warenzeichen von Lotus Dev. Corp. Turbo Pascal und Turbo BASIC sind Warenzeichen von Borland Corp. DBase III Plus ist ein Warenzeichen von Ashton-Tate Corp.

Nähere Informationen gegen ausreichend frankierten Rückumschlag bei:



MERLIN COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL.: 06196/481811

Leser -> ST

Betr.: ST 3/88 "Nichts dabei gedacht?"

Seit fast zwei Jahren benutze ich tagtäglich 1ST(PLUS) in mehreren Versionen. Davor habe ich anderthalb Jahre lang mit WORDSTAR geschrieben, mit einigen anderen der "großen" MS-DOS-Textprogrammen bin ich ebenfalls vertraut. Auf diesem Erfahrungshintergrund kann ich Michael Spehrs Artikel über die 1ST WORDPLUS-Version 2.02 in Heft 3/88 teils zustimmen, teils aber muß ich ihm widersprechen.

Insgesamt halte ich 1ST WORDPLUS für ein außerordentlich klares, sauberes, benutzerfreundliches, in einiger Hinsicht geradezu geniales Programm. Es hat all den großen MS-DOS-Textprogrammen vor allem eine Qualität voraus: Es macht sich selbst kaum bemerkbar - man muß es zum Beispiel nicht erst lernen, sondern beherrscht es sozusagen aus dem Stand. Darum ist es besonders bedauerlich, daß GST und ATARI das Interesse an seiner Weiterentwicklung verloren zu haben scheinen; sonst hätte vielleicht ein in jeder Beziehung optimales Programm aus ihm werden können. Mir jedenfalls wird der Abschied von ihm schwerfallen.

Völlig zu unrecht wird ihm immer wieder vorgeworfen, es sei beim Scrollen im Text "entsetzlich langsam und träge". Tatsächlich kenne ich kein Programm, das es erlaubte, so leicht und schnell selbst durch lange Dateien zu scrollen - mit einigem Augenmaß ist man genauso schnell an der gesuchten Stelle, und sei sie noch so fern, wie es dauert, das weiße Quadrat im Rollbalken rechts auf und ab zu schieben.

Gleichwohl hat es Schwächen. Ich glaube, man tut gut daran, die wenigen echten Fehler (die auf jeden Fall noch beseitigt werden müßten, wenn es weiter eine Rolle spielen will) von den bloßen Wünschbarkeiten zu unterscheiden.

Die Fehler:

1. Das Programm stürzt gelegentlich ab: der Cursor beginnt, im Text hin- und herzuspringen, die Tasten reagieren nicht mehr. Manchmal gelingt es gerade noch, den aktuellen Textstand zu sichern - meist aber ist er verloren.
2. Die Trennhilfe versagt nicht nur, wenn Anführungszeichen in der Zeile stehen, sondern auch bei Klammern und unter einigen anderen, schwer durchschaubaren Bedingungen.
3. Beim Neuformatieren veränderter Textpassagen ignoriert es häufig Carriage Returns, so daß Absätze aneinandergehängt werden. Vielleicht hat es unter bestimmten Bedingungen manche Carriage Returns auch vorher gelöscht - man sieht es ja nicht. Eine Abhilfe wäre es, die Carriage Returns doch mit irgendeinem kleinen Symbol auf dem Bildschirm darzustellen.
4. Weder beim wortweisen Löschen noch bei der Rechtschreibüberprüfung erkennt das Programm getrennte Wörter als solche, sondern behandelt fälschlicherweise beide Hälften, als seien sie eigene Wörter.

Wünschenswert wären einige zusätzliche Funktionen:

1. Die Möglichkeit, den Text in anderthalbzeiligen Abständen ausgeben zu lassen.
2. Blocksatz im Feinausgleich der Wortabstände direkt aus dem Programm, auch in Proportionalschrift.
3. Die Möglichkeit der Spaltenbildung und- formatierung.
4. Die Aufhebung der Längenbeschränkung bei den Fußnoten.
5. Endnoten zusätzlich zu den Fußnoten.
6. Die Möglichkeit, einige Default-Parameter selber einzustellen, vor allem die Trenntiefe und die automatische Seitennumerierung.
7. Die Möglichkeit, Desktop-Funktionen aufzurufen, ohne das Programm zu verlassen.
8. Eine Überschreibfunktion wie etwa bei EUROSCRIPT, die es erlaubte, akzentierte Zeichen zu schreiben, ohne erst zum Sonderzeichenfont zu blättern; oder die Möglichkeit, bestimmte Zeichen aus dem Font auf einzelne Tasten zu legen.

Schön wäre es auch, wenn die deutsche Betextung so geändert würde, daß dem Anwender nicht andauernd dieses falsche Deutsch entgegenkommt.

Das Rechtschreibprüfprogramm hat einen Grundfehler, der nur behoben werden könnte, wenn man noch einmal bei Null anfinge. Nämlich: es ist in seinen Leistungen linguistisch auf die englische Sprache eingestellt, nicht aber auf die anders strukturierte deutsche. Die Wörter der deutschen Sprache sind länger; sie hat ungleich mehr flektierte Formen; sie bildet zusammengesetzte Wörter in großer Zahl. Alles führt dazu, daß ein deutsches Lexikon sehr viel umfangreicher sein muß, damit es Vergleichbares leistet wie ein englisches. Dem schieren Umfang sind hier Grenzen gesetzt: durch die Kapazität des RAM, durch die Ladezeit, durch die Datenmenge, die das Rechtschreibprüfprogramm verwalten kann. Darum muß das Lexikon komprimiert werden. Indem sie ein durchschnittlich 12 Byte-langes deutsches Wort auf 1,5 Byte reduzieren, haben GST und ATARI wohl ein Äußerstes an Datenkompression verwirklicht. Bliebe also nur, den Platzbedarf des Lexikons von vornherein drastisch zu verringern. Dazu gäbe es wohl nur einen Weg: Es müßte Zusammensetzungen seiner einzelnen Einträge als legitime Wörter anerkennen. Dann erübrigte es sich zum Beispiel, alle Verben, die mit auf- beginnen, und das sind Hunderte, einzeln in ihren konjugierten Formen aufzunehmen (durchschnittlich neun bereits, wenn man auch nur die häufigsten berücksichtigt) - die konjugierten Grundverben und die einzelnen Präpositionen würden genügen; auch die zum Teil völlig unvorhersehbaren zusammengesetzten Substantive, die das Deutsche zu reichlich bildet, wären zu einem großen Teil abgedeckt. Dadurch wäre das Lexikon schätzungsweise um über ein Drittel zu reduzieren; und an die Stelle dieses Drittels könnte man viele dringend nötige neue Einträge setzen.

Das alte deutsche Lexikon zu 1ST WORDPLUS hatte nach meiner Berechnung etwa 46.000 Einträge und dabei nur spärliche Flexionsformen. Das neue hat nach dem gleichen Schlüssel 117.000 und zu sämtlichen Lexemen die gebräuchlichsten Flexionsformen (durchschnittlich 4). Für neue wirklich flüssige Arbeit mit ihm wären schätzungsweise 150.000 Einträge erforderlich. Darum sollte sich niemand an dem "Ballast" an Einträgen stören, die er persönlich wahrscheinlich niemals brauchen wird - jedes Lexikon ist nur gut, wenn es voll ist mit Lexemen, die einem selber egal sind. Natürlich wäre es nicht schlecht, wenn man auch im Hauptlexikon Einträge löschen könnte; aber wichtig ist das nicht. Selbst wenn man 1.000 Wörter fände, auf die man selber gerne verzichtete, wie auf das Wort "mausetot" - ihre Löschung würde die Bearbeitungszeit nicht beschleunigen und die Ladezeit (von der Festplatte) nur um zwei Drittelsekunden verkürzen. Man sollte sich also besser um die Erweiterung statt um die Verkleinerung des Grundlexikons Sorgen machen. Gut 30.000 Einträge brauchte es noch. Das wäre einiges an zusätzlicher Arbeit (mindestens 100 Arbeitsstunden). Aber auch wenn sich jemand die Mühe machte: Es wäre vor allem nicht gesagt, daß das Prüfprogramm ein so umfangreiches Lexikon noch verwalten könnte. Ich konnte von ATARI jedenfalls nicht erfahren, welches denn nun die Grenze ist. Da liegt der Hase im Pfeffer.

Um das Rechtschreiblexikon als Ausnahmewörterbuch für die Silbentrennung zu benutzen, müßte es ebenfalls von Grund auf neu gebaut werden - das wird also wohl nie geschehen. Aber die Trennhilfe ließe sich auf eine sehr viel einfachere Weise wesentlich verbessern: Sie müßte nur statt des englischen einen deutschen Trennalgorithmus bekommen.

Es hat meiner Meinung nach wenig Sinn, GST und ATARI jetzt an den Kopf zu werfen, 1ST WORDPLUS wäre nur noch dann "zu retten", wenn sie Änderungen vornehmen, die auf eine völlige Neukonzipierung ganzer Programmteile hinausliefen. Sehe ich recht, wird es dazu bestimmt nicht kommen. Um so deutlicher aber müßte man darauf dringen, daß wenigstens die verbliebenen Bugs beseitigt und vielleicht noch ein paar weniger grundlegende Veränderungen vorgenommen werden. So wäre ohne allzu großen Aufwand doch noch eine sehr wesentliche Verbesserung zu erreichen.

Dieter E.Zimmer, Hamburg

Betr.: ST 4/88 "AnsiTerm"

In dem von Ihnen veröffentlichten Testbericht über das Programm AnsiTerm haben sich leider zwei Fehler eingeschlichen.

1. In dem Artikel heißt es wörtlich: "Leider ist nicht vorgesehen, die empfangenen Daten auf Diskette zu sichern, und...". Dies ist falsch, wie man aus dem daneben abgebildeten Menü ersehen kann (Bild 1), gibt es die Funktion "File capture", die die empfangenen Daten in einer Datei abspeichert.
2. Auf Seite 116 behauptet der Autor, daß die MT-C-Shell 800k im Speicher belegt. In Wirklichkeit belegt sie nur ca. 250k. Richtig ist vielmehr, daß bei einer Verwendung der MT-C-Shell und dem VSH-Manager auf einem 1 MB-ST ca. 200k freier Speicherplatz übrig bleiben.

G.Sender, Computerware

ENDE

Anmerkung der Redaktion über den Beitrag "Drucker Scanner im Selbstbau"

Das abgedruckte Listing ist eine abgespeckte Version von dem Original das von der Public Domain Sammlung zu beziehen ist. Eine Fülle von Funktionen sind noch nicht immplementiert. Das gesammte Programm abzudrucken hätte noch einige Seiten mehr in Anspruch gennomen. Weiterhin enhält das Listing einige Fehler:

Zeile Nr.	steht	soll
8	Urbild$=Space$(32767)	Urbild$=Space$(32766)
9	Bild$=Space$(32767)	Bild$=Space$(32766)
961	If W(34)*W(35)/2>32767	If W(34)*W(35)/2>32766

Immer up to date

Mit dieser Sparte wollen wir allen unseren Lesern die Möglichkeit geben, sich über die neuesten Programm-Versionen zu informieren. Angegeben werden die aktuelle Versionsnummer, ein eventueller Kopierschutz, die Bildschirmauflösungen und der Speicherbedarf. Softwarefirmen ist es somit möglich, die ST-Computer-Leser über ihre Updates zu informieren.

Programmname	Version	Daten	Programmname	Version	Daten
AnsiTerm	1.1	N	1st_Lektor	1.2	H
Binlook	1.0	N HML	Lisp Complete	1.01	N HM
BTX für ST	1.0	N H 1M	Lock_it	1.0	J ML
Crypt_it	1.0	J HML	Megamax C	1.1	N HML
Flash-Cache/Flash-Bak	1.0	N HM	Mega Paint	1.0	N H 1M
Flexdisk	1.2	N HML	Megamax Modula 2	1.0	N HM
1st_Freezer	1.0	N HML 1M	Micro C-Shell	2.70	N HM
GFA-Artist	1.0	N L	MT C-Shell	1.10	N HM 1M
GFA-BASIC 68881	1.3	N HML	Multi ST	1.0	N HML 1M
GFA-Compiler	2.02	N HML	Musix32	1.01	J H
GFA-Draft	2.1	N	Omikron BASIC Compiler	2.0	N HML
GFA-Draft plus	1.5	N	Omikron BASIC Interpreter	2.0b	N HML
GFA-Interpreter	2.02	N HML	PC ditto Euroversion	3.64	N HML
GFA-Objekt	1.1	N HM	Pro Sound Designer	1.2	L
GFA-Starter	1.1	N HML	Pro Sprite Designer	1.0	L
GFA-Vektor	1.0	N	Search!	1.31	N HM
Hard Disk Accelerator	1.0	N HML	Signum! zwei	2.0	N H
Hard Disk Toolkit	1.04	N HM	1st_Speeder	1.01	N HML
Harddisk Utility	1.04	N HM	1st_Speeder 2	1.0	N HML 1M
Imagic	1.0	N HML	STAD	1.2	N H
Intelligent Spooler	1.01	N HML	ST Pascal plus	2.02	N HM
K-Resource	1.1	N HM	Tempus	2.0	N HM
Laser Deluxe	1.0	N HML 1M	VSH Manager	1.11	N HML 1M

Irrtum vorbehalten
Daten-Legende : N = kein Kopierschutz, J = Kopierschutz, H = hohe Auflösung, M = mittlere Auflösung, L = niedrige Auflösung,
1M = mindestens 1 Megabyte Speicher notwendig

ST_COMPUTER PUBLIC DOMAIN KLASSIKER

Unser Redaktionsservice für unsere Leser.

Die nicht aufgeführten Disketten sind natürlich auch weiterhin erhältlich, nur wurde uns der Platz zu klein. Schauen sie dazu bitte in einer der vorherigen Ausgaben nach oder fordern Sie die Liste an (frank. Rückumschlag).

Diskette 7
- **XLISP 1.71:** (die Sprache der künstliche Intelligenz) incl. 54 KByte Dokumentation, Anleitung und Beispielen.

Diskette 11
- **PROLOG 10:** Die Sprache der 5. Genertion. Unterstützt GEM.incl. 95 KByte Anleitung und Beispiele.

Diskette 23
- **Das große deutsche Ballerspiel:** besonderes Aktionspiel (s/w)

Diskette 34
- **Assemblerpaket:** nach M 68000-Standard (2-Pass Assembler, Disassembler, Debugger, Editor)

Diskette 40
- **Deluxe Fontmaster ST:** Druckprogramm für Karten oder schöne Tabellen. Viele Zeichensätze, Proportionalschrift und Blocksatz, eigener Zeichensatzeditor.

Diskette 75
- Randvoll mit Zeichensätzen für Fontmaster (Disk 40).

Diskette 49, 50, 51
- Volksforth 83: Die drei Disketten beinhalten ein komplettes Forth-System für den ST. Alle Diskette werden benötigt.

Diskette 52
- **Molekül:** Darstellung chemischer Moleküle. In s/w und Farbe, wobei in Farbe mit 3D-Effekt. Atom, Ionen oder Van der Waalsradius. Ein Molekül kann gedreht und sogar animiert werden.

Diskette 61
- **Hacomini :** Erzeugt Hardcopies in Miniaturformat (48mm * 68mm) (s/w).
- **Helper:** Diskmonitor und Druckerutility.
- **Bidipo:** Scrollt bis über 100 Bilder (4MB) über den Schirm. Für Diashows.
- **Datobert:** Grafische Darstellung. Linien- und Kuchengrafik, sowie in dreidimensionaler Balken- und Flächengrafik. (s/w)
- **Direct all:** Directorydruck.
- **Überweisung:** Überweisungsdruckprogramm.

Diskette 70
- **Orbit:** Satellitenbahnberechnung. Animation über der Weltkarte (s/w)
- **High End:** Programm zum Berechnen von Lautsprecherboxen aller Art. Lautsprecher und Boxenmaße. Datei für Speaker enthalten. (s/w)

Diskette 77
Utilitydisk
- **FCOPY 2.0:** Schnelles Kopierprogramm. Formatieren, Verify, Multiple Copy, Scan und Teilkopie.
- **Speeder:** Verdoppelt Schreib- und Lesegeschwindigkeit.
- **Filecopy:** Kopieren von Einzeldateien. Auch mit einer Floppy.
- **Formater:** Erhöht die Diskkapazität
- **Fastformat:** Formatiert Schnelladedisks.

Diskette 81
Malprogramm
- **Public Painter:** Superzeichenprogramm. Gepacktes Format, SNAPSHOT-Bilder, Lasso, Blockmanipulation, Vergrößern, Verkleinern, Drehen, Biegen, Verzerren, Überschneiden. Drucken von Bildausschnitten.(s/w)

Diskette 86
- **CARPET:** Berechnungsprogramm von 3D-Funktionen, Hidden-Line Algorithmus und Beleuchtungseffekten. Funktionseingabe und frei einstellbare Parameter. (s/w)
- **3D-PLOT:** Ähnliches Programm in compiliertem GFA-Basic, mit einigen Extra-Funktionen. (s/w)
- **Funktionsplot:** Funktionseingabe im Programm, verschiebbare Achsen, mehrere Darstellungsarten (s/w)

Diskette 88
Druckertreiber
- **NEC-EMU:** Hardcopytreiber für NEC P5/P6/P7. Erzeugt Hardcopies in 12 verschiedenen Größen und versch. Kontrasten. Enthält Druckerspooler.
- **LQ 800:** Hardcopytreiber für EPSON 24 Nadeldrucker in drei verschiedene Größen.
- **NEC_CODE:** Ein Programm zur Einstellung des NEC P6 /P7 u.ä..
- **SIMPLE:** Komfortables Hardcopy-Programm mit vielen Extras.
- **STX-80:** Hardcopytreiber für diesen Thermodrucker.
- **WORDPLUSTREIBER:** Druckertreibersammlung für folgende Drucker: NL10, FX1000, SP1000I.

Diskette 90
Utilities
- **MAXIDISK 4:** Die komprimierende Maxidisk mit max. 4MByte Kapazität. Läuft auf ALLEN TOS-Versionen. Komprimiert automatisch ihren Inhalt. RESETFEST.
- **TOPS:** The Other Pascal Shell (für PascalPlus), TEMPUSfähig, erzeugt Crossreferenz. Kopieren, Löschen und Umbenennen von Files.
- **SIGNUMSHELL:** Eine Shell zu Signum. Umschaltung zwischen Signum Editor, den verschiedenen Druckprogrammen, den Zeicheneditoren per Menüleiste. Automatisches Install.
- **TEMPELMON 1.8:** DER weitverbreitete Monitor.
- **RAMDISK:** Gleiches Programm wie MAXIDISK, nur ohne Komprimieralgorithmus.
- **COPY:** Beide Ramdisk enthalten ein Autocopyprogramm zum kopieren bestimmter Files in Ramdisk.
- **RCS.WANDEL:** Wandelt *.h Files nach Basic und Assembler.

Diskette 93
- **ADR_2:** Semiprofessionelles Adressverwaltungsprogramm mit Serienbrieferstellung (mit und ohne 1st Word). Edierbare Druckeranpassung (Steuerzeichen, Zeichenwandlung). erzeugt Adress- Telefon und Geburtstagslisten. Druckt Labels, Etiketten nach freidefinierbarem Format. (s/w)
- **DAT_TEXT:** Adressverwaltung für Mehrpersonenzugriff auf eine Datei. Datenzugriff nur durch Passwort (s/w)

Diskette 100
- **ST-SPEECH:** Sprachausgabe auf Softwarebasis. Wandelt englischsprachige Texte automatisch in verständliche Lautschrift. (s/w)
- **FORTRAN SHELL:** Eine Shell für Fortran 77. Bequemes Aufrufen von Compiler, Linker und anderen Programmen.
- **LIFE IS LIFE:** Life-Simultionsprogramm. Sechs verschiedene Feldgrößen, Drehen, Verschieben, Kopieren von Blöcken wie im Malprogramm, Festlegen der Fortpflanzungsregeln, hohe Geschwindigkeit. (s/w)
- **FASTLIFE:** Life-Simulation in Volksforth. Wahnsinnige Geschwindigkeit. (s/w)
- **GEMFRAC:** Berechnung fraktaler Landschaften in Farbe und s/w.
- **SHARP:** Basic Programme für Sharp-Basicrechner. Geeignet für Direktüberspielung mit Interface.

Diskette 113
- **ZEITMANAGER:** Semiprofessionelle Terminplanverwaltung. Eigener Desktop, Terminerinnerung, Listendruck, Timingverwaltung, fixe Termine, Kalender. Schaltjahrfest (s/w)

Diskette 114
Spiele
- **METROPOL:** Wirtschaftsspiel. Retten Sie die Wirtschaft ihres Landes. 'Auf der Suche nach der Wende'.(s/w)
- **DALLAS:** Öl, Öl, Öl, Geld, Geld, Geld, JR, JR, JR (f)

Diskette 115
Spiele
- **SHERLOCK:** Klären Sie als Holmes den Mord.
- **EL BOZO CITY:** Textadventure mit vielen Gags.
- **SCRIBBLE:** Schöne Scrabble-Version für den ST. (s/w)
- **KREUZWORT:** Generierung eines Kreuzworträtsels. (s/w)

Diskette 116
- **µEMACS 3.8:** Deutsche Version. Eigene Macrosprache, Textverschlüsselung, Mailmerge, mehrere Screens, u.v.a.m.

Diskette 117
Terminalprogramme
- **UNITERM:** Exclusives Terminalprogramm. VT200, VT102, VT100, VT52, 4010, u.a. Softscroll, Grafikübertragung, Funktionstastenbelegung,..

Diskette 118
Utilities
- **M_COPY:** Universelles Kopierprogramm bzw. Diskmonitor.
- **LOADER:** Aktiviert oder deaktiviert Accessories bzw. Autostartprogramme.
- **VIEW:** Programm zum Anzeigen von ASCII-Files. Vor- und Rückblättern per Tastendruck.
- **SUPERSHELL:** Universelle Shell. Aufruf beliebiger Programme aus der Menüleiste. Source in GFA-Basic.
- **LATTICE:** diverse Utilities: HEXDUMP, Ausdruck mit Zeilennummern, Symboltabellen, Archivierung und Pflege von Modulbibliotheken.

Diskette 119
Etiketten Drucker
- **ST Label:** Etikettendrucker
- **LABEL:** Komfortabler Disketikettendrucker. Übernahme eines Bildes per Mau 'ick. Eigene Bilder integrierbar. (s/w)
- **PASTE ADRESS:** Adressaufkleber in Miniformat (1.5cm*2.5cm)

Diskette 120
- **WISSEN SIE ES?:** Quizspiel mit vielen Fragen aus verschiedenen Gebieten. Ähnlich 'Trivial Pursuit'. Viele Fragen aus verschiedenen Wissensgebieten, eigene integrierbar. (s/w)

Diskette 121
Spiele
- **GO_UP:** Loderunner-ähnliches Spiel. Auf vielen Mauern, Leitern und Seilen müssen Sie den Verfolgern entkommen. Viele Level. Editor zum Entwickeln eigener Levels enthalten. (s/w)
- **MAZE_EDIT:** Editor zu dem Spiel 'MIDI MAZE'.
- **F1 MANAGER:** Verwaltung eines Formel-Eins-Rennstall. (s/w)
- **OELIMPERIUM:** Managen einer Ölfirma. (s/w)

Diskette 122
Spiel
- **PD_BOLO:** Sie kennen Arkanoid, dann sollten Sie auch PD_Bolo kennen. Interessante Break-Out Variante. (mind. 1 MByte)

Diskette 123
Utilities
- **Hyperformat:** Formatierprogramm von Claus Brod.
- **FSELECT** Die neue Fileselectorbox. Wird fest installiert. Kommt bei jedem Aufruf.Warum nicht gleich so, ATARI?
- **Diskmon:** Diskettenmonitor. (s/w)
- **PC_Ass:** Assembler für Sharp-Taschencomputer.

Diskette 124
Chemie
CHEMLIB Dateiverwaltung für chemische Elemente (s/w)

Diskette 125
Zeichenprogramm
- **CAD 2:** Objektorientiertes Grafikprogramm, z.B. mit Zoom ohne Auflösungsverlust.(s/w)

Diskette 126
- **CUBE HACK:** Terminalprogramm mit integrierter, umfangreicher Programmiersprache. Anwendungen: z.B. Automatisches Einloggen in Mailboxen.
- **OR Spread:** Spreadsheetähnliches Programm zur Linearoptimierung, Netzplanentwicklung, kürzeste Wege Berechnung. (s/w)

Diskette 127
Wissenschaftliche Anwendungen
- **Planet:** Himmelskörperberechnungen. (Grafische Darstellung des Sternenhimmels, Horizont). Animation.
- **Turing:** Realisierung eines Turing-Maschinen-Modells (s/w)
- **Zust:** Analyse und Simulation linearer Regelkreise (s/w)

Diskette 128
Spielesammlung
- **Horrorschloß:** Der Weg durch viele Gänge birgt viele Gefahren.
- **Inversi:** Das bekannte Spiel, gegen den Rechner oder zu zweit (s/w)
- **Invasion:** Space Invadors mit 3-D Vektorgrafik. (s/w)
- **Zarge:** Ein 32000K-Bild wird gepuzzelt (s/w)

Diskette 129
Utilities
- **GFA-Shell:** Shell für GFA-Basic, incl. CLI und Sprite-Editor (s/w)
- **OMIKRON-Shell:** Shell für OMIKRON.-Basic System. Aufruf von Interpreter, Compiler, Linker. Generiert PRG, TOS, TTP oder ACC. (s/w)
- **Disksave:** Speichert Bildschirm per 'Alt+Help' auf Disk.
- **Apple-ST:** Datenübertragung Apple II—> ATARI ST.

Diskette 130
Utilities
- **HFORMAT:** Formatierprogramm als Accessory.
- **Rescue:** Rettet gelöschte Files. (s/w)
- **Diskinfo:** Gibt Directory sortiert auf Drucker (Liste oder Label) bzw. Datei aus. (s/w)
- **AccLoader:** Ermöglich die Auswahl bei mehr als 6 Accessories (s/w)

Diskette 131
Macros zu VIP-Professional (professionelle Beispiele zum VIP-Kurs im Heft) (1MByte Ram)
- **Verein:** vollständige Vereinsverwaltung. Serienbriefe, Lastschrifteinzug, Überweisungen, Etikettendruck, Listendruck.
- **Meister:** komplette Verwaltung einer Vereinsmeisterschaft.

Diskette 132
Simulationen
- **Water:** Simulation eines biologischen Systems am Zusammenleben von Haien und Fischen.
- **Life:** Ein sehr flexible Life-Version. (s/w)
- **Quicklife:** Rasend schnelles Life. (s/w)

Diskette 133
- **Codewars:** REDCODE-Programm. Der Kampf im Rechner. Incl. ausführlich Einführung in die Theorie.
- **Kerne 2.1:** Erweiterte Version der 'Krieg der Kerne'.

Diskette 134
Spielesammlung
- **Galerie:** Wunderschöne Patience-Version. Für kalte Frühlingsabende.
- **Kreuz As:** Umsetzung des Spielautomaten. Mut zum Risiko ist angesagt. (s/w)

Diskette 144
- **Midi-Musik:** Komfortables Musik-Programm. Eingebauter Notendruck (s/w)

Zeichenerklärung:

(s/w)	*- Nur Monochrom*
(f)	*- Nur Farbe*
kein Kürzel	*- Farbe und Monochrom*

Sonderdisks

Die folgenden Programme sind nicht Public-Domain.
Sie können aber bei uns bezogen werden.

A.) TOS: Die letzte Disketten-Version vom 6.2.1986. Ältere Versionen laufen nicht problemlos.
Unkostenbeitrag DM 15.-

B.) RCS: Das Resource-Construction Set aus dem ATARI-Entwicklungspakets. Unverzichtbar bei der GEM-Programmierung.
Unkostenbeitrag DM 15.-

Bis auf die Höhe des Unkostenbeitrages gelten die gleichen Versandbedingungen wie bei der Public-Domain Software.

VERSANDBEDINGUNGEN

Sämtliche Disketten können ab dem Erstverkaufstag der ST-Computer direkt bei der Redaktion bezogen werden. Wir haben für Sie den schnellstmöglichen Versandservice eingerichtet. Lieferung innerhalb einer Woche.

1. Schriftliche Bestellung
- Der Unkostenbeitrag für eine Diskette beträgt DM 10.-
- Bezahlung nur per Scheck oder Nachnahme (Im Ausland nur Vorauskasse möglich)
- zuzüglich DM 5,00Versandkosten (Ausland DM 10.-)
- bei Nachnahme zuzüglich DM 3.70 Nachnahmegebühr
- Legen Sie bitte, falls zur Hand, einen Aufkleber mit Ihrer Adresse bei.
- Bitte fügen Sie keine anderen Bestellungen oder Anfragen bei.

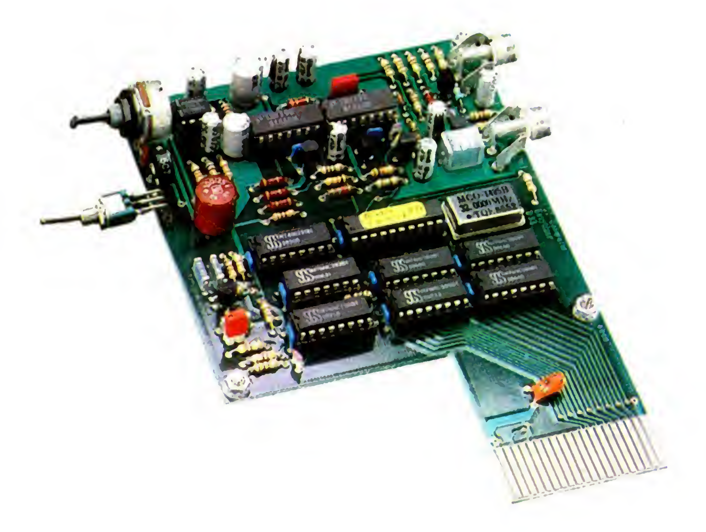
2. Anruf genügt
'MERLIN'-Computer GmbH
ST-Computer Redaktion
Tel.: 0 61 96 / 48 18 11
Von Mo-Fr 9.00 bis 17.00 Uhr

Die Bezugsadresse lautet:
'MERLIN'-Computer GmbH
ST-Computer Redaktion
'PD-Service'
Postfach 5969
D-6236 Eschborn

Bei Fragen bezüglich der Programme stehen wir Ihnen gerne zur Verfügung.

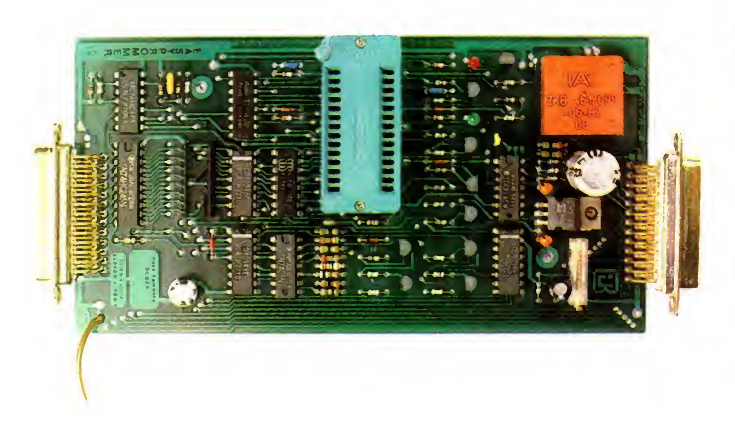
Aus der Hardwareküche





Der Easyprommer V2.2

Ein universelles Programmiergerät für den ATARI ST, das nicht nur alle gängigen EPROMs der 27...-Serie inkl. 27011 (Megabit-Eprom) brennt, sondern auch die modernen EEPROMs lesen und programmieren kann. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen. Mit dem integrierten HEX/ASCII-Monitor lassen sich Änderungen der Daten blitzschnell realisieren. Der auf der Platine befindliche Druckertreiber erlaubt das problemlose Drucken der Daten. Zwei Zusatzprogramme ermöglichen das Brennen **jeder** nicht kopiergeschützten Software direkt von Diskette einschließlich Accessories und Autostart-Programmen. Der Easyprommer wird an den Druckerport angeschlossen und samt Diskette und Handbuch geliefert.



Easytizer – der Videodigitizer ohne Geheimnisse

Mit dem Easytizer können Sie beliebige Videosignale von einer Schwarzweiß- oder Farb-Kamera, Videorecorder oder auch direkt vom Fernsehgerät (mit Composite Video Ausgang) digitalisieren und somit auf dem Bildschirm Ihres ATARI ST sichtbar machen.

Der Easytizer wird am Modul-Port des ST angeschlossen. Die Auflösung beträgt 800x600 Bildpunkte, so daß in mittlerer Auflösung 640x200 Bildpunkte in vier Graustufen dargestellt werden können. In dieser Betriebsart werden **12,5 Bilder in der Sekunde** wiedergegeben! Im hochauflösenden Modus werden 640x400 Bildpunkte in Schwarzweiß wiedergegeben.

Besondere Merkmale des Easytizers:

- Software vollständig in Assembler
- Abspeichern der Bilder im DEGAS-Format
- Spiegeln eines Bildes in horizontaler und vertikaler Ebene
- Von STAD ansteuerbar
- Ein beliebiger Bildausschnitt kann in ein mit der Maus wählbares Format verkleinert oder vergrößert werden
- Animation mit beliebig vielen Bildern möglich, nur durch die Kapazität des Rechners begrenzt (beim Mega ST4 über 100 Bilder)
- Eingebauter Druckertreiber für die mittlere Auflösung für NEC P6/P7 und EPSON oder Kompatible
- Wahlweise automatische oder manuelle Helligkeitseinstellung
- Schnappschuß

Lieferumfang:

1.) **Fertiggerät**
komplett aufgebaut und geprüft, inklusive Diskette mit der Easytizer Software und Bedienungsanleitung. Preis: DM 289,–

2.) **Teilesatz**
Für Bastler liefern wir einen Teilesatz bestehend aus:
- doppelseitiger, elektronisch geprüfter Platine mit Lötstopplack und Bestückungsaufdruck sowie vergoldeten Anschlußkontakten
- fertig programmiertes GAL 16V8
- Quarzoszillatormodul 32 MHz
- Diskette und Bedienungsanleitung — Preis: DM 129,–

Hiermit bestelle ich

AUSLAND NUR GEGEN VORKASSE

☐ Easyprommer (Fertiggerät) für DM 349,–
☐ Easyprommer (Bausatz) für DM 298,–
☐ Easytizer (Fertiggerät) für DM 289,–
☐ Easytizer (Teilesatz) (w. o. beschrieben) für DM 129,–
☐ 128 KByte EPROM-Karte (fertig bestückt o. EPROMs) für DM 58,–
☐ Verlängerungskabel für Easyprommer für DM 39,90

Versandkosten: Inland DM 7,50 Ausland DM 10,–
Auslandsbestellungen **nur** gegen Vorauskasse

☐ Vorauskasse
☐ Nachnahme

Name: ______ Vorname: ______

Straße: ______ Ort: ______

Unterschrift: ______

Bezugsquelle:



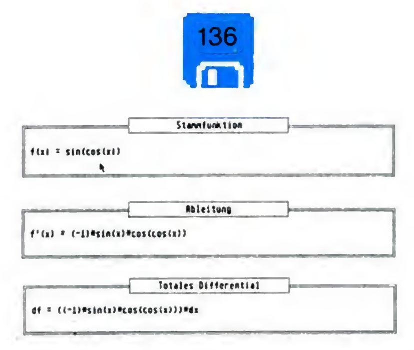
ST-COMPUTER PUBLIC DOMAIN SERVICE

Liebe Leser,

seit über zwei Jahren gibt es nun schon unseren Public-Domain-Service. Eine lange Zeit, nach der man vermuten könnte, daß es nun zu jeder Anwendung ein PD-Programm gibt. Doch wie bei der käuflichen Software, gibt es auch bei den PD-Programmierern ständig neue Ideen und perfektere Programme. So auch in diesem Monat.
Einen kleinen Rückblick (unsere Empfehlung sozusagen) über die letzten zwei Jahre haben wir für Sie auf der rechten Seite zusammengestellt. Darunter sind viele bisher unübertroffene Klassiker, die wir wärmstens empfehlen können.
Viel Spaß

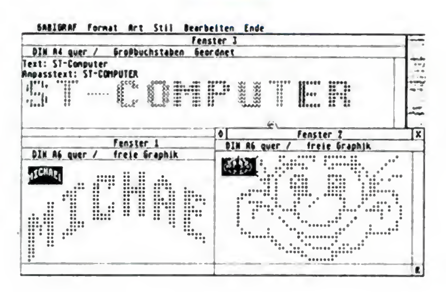
Ihre ST-Computer Redaktion

Neu in diesem Monat



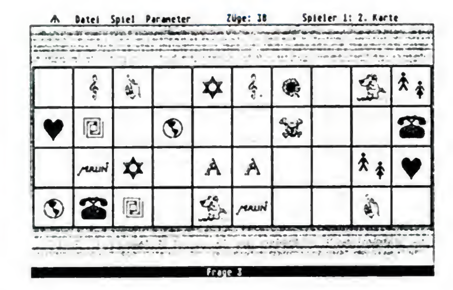
IQ-Test: Gibt nach vielen grafischen und mathematischen Tests Ihren errechneten IQ-Faktor an. (s/w)
GENIUS: Programm zur Diagnostizierung des physischen Leistungsstandes. Reaktionstests, Gedächtnisschulung. Auswertung der Ergebnisse. (s/w)
DIFFERENTIAL: Von einer Stammfunktion wird die Ableitung, das totale und das partielle Differential berechnet. (f(x), f(x,y), f(x,y,z)). Ausgabe als Funktionsgleichung. (s/w)
KLIMA: Grafische Ausgabe von Klimatabellen der ganzen Welt. Eigene Eingaben und Auswertungen möglich.
ZAHN: Stirnrad-Vorauslegungsberechnung. Hoffnung für Maschinenbaustudenten. (s/w)



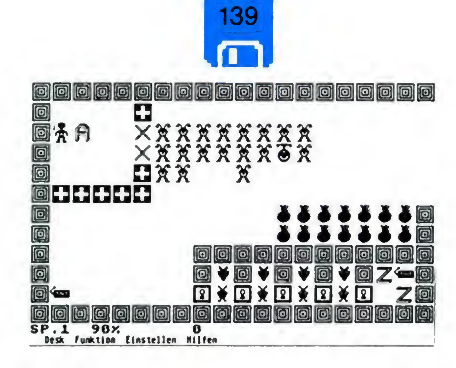


Druckertreiber
STAR LC10: Verschiedene WORDPLUS-Treiber für den neuen Low-Cost-Drucker.
STAR NL10: Altbekannt aber dennoch rüstig. Mehrere WORDPLUS-Treiber mit verschiedenen Optionen.
CITIZEN: WORDPLUS-Treiber für 120D, LSP 10 und LSP 11.
GABIGRAF: Grafiktreiber für Typenradmaschine und Matrixdrucker. Gibt Grafik und vielseitig gestaltbaren Text aus (siehe Bild).





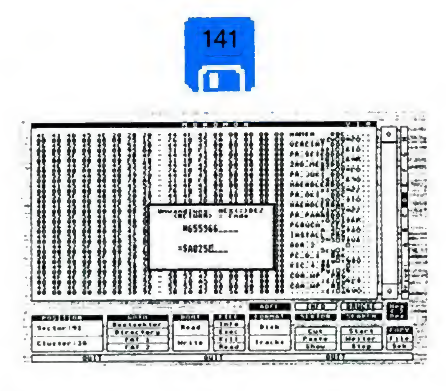
MEMORY: Umsetzung des bekannten Brettspiels. Mehrere Kartenpakete enthalten. Eigene Spiele können auf einfachste Weise mit jedem Zeichen- oder Malprogramm erstellt werden. Feldgröße und Anzahl der Karten frei wählbar. Dadurch anwendbar für Kinder und Gedächnisgenies. (s/w)



DGDB Das Große Deutsche Ballerspiel nun endlich auch in monochromer Version (Farbversion auf PD23). Aktionspiel. Kreuz und quer durch Gänge mit viel bösem Getier. Alleine oder zu zweit spielbar. (s/w)
SCHLOSS Textadventure mit Sprachausgabe. Als Sohn des Pink-Häuptling Tu Wat und dessen Frau Mach Wat machst Du Dich auf den Weg zu dem unheimlichen Schloß des großen Zauberers, um von Ihm Hilfe gegen die Überfälle der Puwacks zu erhalten. Doch im Schloß warten viele Gefahren.



Haushaltsverwaltung
ETAT: Planung und Kontrolle des eigenen Haushalts. Bestehend aus 5 Teil-Programmen: Buchung, Monatsplanung, Kontenverwaltung, Jahresplanung, Jahresabrechnung, die über Menüführung verbunden sind. Bringt die Haus-Kasse unter Kontrolle. (s/w)



Utilities
COMMAND: Commando-Interpreter mit folgenden Extras: Dateien packen/entpacken, verschlüsseln, Funktionstastenbelegung, Find, Passwortabfrage, History-Funktionen, Zahlenumwandlung, Batchprozessing (Loop, Goto, Vergleichsoperationen, Variablen)
MONOMON: Monitor für Floppy, Festplatte und Ramdisk. Ganzer Sektor auf Bildschirm, Direktanfahren von Bootsektor Directory und Fat. Bootsektor erstellen. Cut und Paste. PRG's ausführen, Schnelles Suchen, u.v.a.m. (s/w)
VIRUSKILL: Programm zum Zerstören von VIREN und regenerieren der betroffenen Disketten.
NOBACKUP: Löscht *.BAK-Dateien
HYP-COPY: Kopierprogramm für Hyperformat-Disketten.
S-FORMAT: Formatieren einzelner oder mehrerer Sektoren. (s/w)
TOTALDELETE: Physikalisches Löschen von Files.

Floppy-Umbau: 259,-
Atari 1040-Reparatur
(Laufwerk)
bitte nachfragen
ST-16 C3½" + 5¼"
40/80 Tracks: 699,-

FISCHER
COMPUTERSYSTEME
Goethestraße 7 · 6101 Fränkisch-Crumbach
Telefon (06164) 4601

---

wissenschaftliche
STATISTIK
mit
WiSTat

einfach in der Anwendung
vielseitig in der Leistung

- → alle einfachen Testverfahren (t-Tests usw.)
- → mehrfaktorielle Varianzanalysen
- → Korrelationen, multiple Regression
- → Faktoren-, Cluster-, Regressionsanalyse
- → Item- und Diskriminanzanalyse
- → Bequemes Dateienhandling
- → Umfangreiche Datentransformationen
- → Ausgabe aller Ergebnisse auch auf Drucker
- → Hotline, Update Service, Sonderwünsche...

wir scheuen keinen Vergleich!

mit Lehr-Handbuch nur noch 398, – DM
12-seitiges Info:
Thomas Leschner ☎ **0641/47459**
Grünberger Straße 81 **6300 Gießen**

T H E A R T O F S O F T W A R E



- SCHNELLER DYNAMO COMPILER
- KOMPLETTES SYSTEM MIT SIMULATOR UND DISPLAYER
- SOURCE KOMPATIBEL ZU PROFESSIONAL DYNAMO
- LÄUFT UNTER GEM, HIRES, FARBE UND IN DEUTSCH
- RERUNS MÖGLICH, BATCH-FILE STEUERUNG
- VIELE EINGEBAUTE FUNKTIONEN
- GRAFIKAUSAUSGABE AUF MONITOR UND DRUCKER
- DEUTSCHES HANDBUCH MIT REFERENZKARTE
- MS DOS VERSION AUF ANFRAGE

DM 225,-

TAOS · P. EXL · P.O. 1002 · 7140 LUDWIGSBURG · 07141-57858

COMPY SHOP

Wir haben die neueste Software für Ihren ATARI ST!

Zum Beispiel:	DM
Carrier Command	79,90
WIZWAL (Dt.)	69,90
DIZZY WIZZARD (Dt.)	69,90
FOOTBALL MANAGER	49,90
SPACE BALLER (Dt.)	29,90
SPACE QUEST 2	69,90
CHECKMATE (Dt.)	34,90
XNON	34,90
EYE (Dt.)	44,90
KING QUEST alle drei Teile zusammen	79,90

Und vieles mehr! Fordern Sie unsere kostenlose ST-Software-Liste an!

COMPY-SHOP
Gneisenaustraße 29 · 4330 Mülheim Ruhr
Telefon (0208) 497169

COPY-POWER
WITHOUT THE PRICE :
REPLICA BOX ST !

Replica-Box fängt da an, wo der WD 1772 aufgibt! Überlange oder verschobene Tracks, CRC-Errors, damaged sectors u.s.w., so heißen die Schreckgespenster von gestern!
Jetzt können Sie Ihre Originale beruhigt in den Schrank legen...

Digital Works * Brünebrede 17
4410 Warendorf * 02581/61126
nur 249.-DM (plus Versandkosten)

PRINT TECHNIK
8000 München 40 · Nikolaistr. 2 · Tel. 089/368197 · Telex 523203 d

PRINT-TECHNIK ATARI UNIVERSAL-SCANNER DM 1.998,-
Print Technik hat einen eigenen Flachbett DIN A4-Scanner entwickelt Preis inclusive TIMEWORKS Desktop Publisher!!!
Der Scanner kann gleichzeitig als Bilderfassungsgerät, Kopierer und Drucker eingesetzt werden. Druckdichte 200 dpi. Läuft auf 320x200, 640x200, 640x400. Scanntime 10 sec. kommt mit ganzseitigen Malprogramm „ROGER PAINT" Speichern der Bilder in allen Formaten möglich. Thermodruck mit GEM-PAINT etc. möglich.

OCR SOFTWARE ... DM 698,-

VIDEO DIGITIZER PRO 8805
Auflösung bis zu 1024x512 + 128 grau. Langsamer hochauflösender Digitizer für professionelle Anwendung ... **DM 498,-**

VIDEO DIGITIZER REALTIZER
Schneller Digitizer für 320x200 und 640x400 unterstützend ... **DM 248,-**
(Beide Digitizer unterstützen alle gängigen Zeichenformate und Desktoppublisher sowie verfügen über ein Tool zum Verändern des Bildes. Kompatibel mit s/w und Colorkamera sowie VCR.

VIDEO-TEXT-EMPFANGS-MODUL
Dieses Modul erlaubt in Verbindung mit der Software den VIDEO TEXT Ihres Fernsehers oder VCR's auf dem Bildschirm des ATARI darzustellen, auszudrucken und abzuspeichern. Empfängt alle Programme auch Sky Channel und Kabelprogramme ... **DM 398,-**

SPEICHERSCOPE ... DM 898,-

KOMPLETTE METEO-SAT EMPFANGSANLAGE DM 3.498,-

Demodisk: DM 15,– Katalog anfordern! (DM 3,–) Täglich Versand

ÖSTERREICH · 1060 WIEN · STUMPERGASSE 34 · TEL. 0222/5973423 · TELEX 112996

SCHWEIZ · MICROTRON · 2542 PIETERLEN · BAHNHOFSTR. 2 · TEL. 032/872429

NL: SOFTPAQUET 079-412563

---

iks Public Domain

Angebot wie 'ST-Computer'
Einzeldisk SS ab 6 Stück **DM 5.00**
bis 5 Stück **DM 6.00**
Doppeldisk z.B 1/2...81/82 u.s.w. (aufeinanderfolgend und ungerade beginnend **DM 7.00**
Pakete 1-10, 11-20, 21-30, 31-40 u.s.w.
auf 5 Disketten DS **DM 30.00**
Preis inkl. Disk zzgl. Porto u. Verp. DM 5.00 (Ausl. DM 10.00) NN plus 1.70, besser V.-Scheck

MS DOS Emulator
Haben Sie Probleme Ihre 5,25" MS Dos Software auf das 3,5" Format zu bringen. Wir lösen das für Sie! Wie? Rufen Sie uns an! Preis **DM 15.00** ab 6 St. **DM 12.00**, ab 10 St. **DM 10.00** je Disk

iks
Schönblickstraße 7
7516 Karlsbad 4
☎ ab 18 Uhr 07202/6793

GEM plus

Das neue Accessory, welches das Betriebssystem des ST um viele Funktionen ergänzt und verbessert:

- Erweiterte Datei-Auswahlbox für alle Programme.
- Dynamischer Maustreiber einstellbar von 25—200% (wie beim Macintosh).
- Hardcopy-Funktion auf Diskette (z.B. für SIGNUM2).
- und viele weitere Funktionen.

DM 59,— zuzüglich DM 5,— Versandkosten

Handy Scanner ST

der leistungsfähige und preiswerte Scanner jetzt endlich auch für den ST. Auflösung 200 dpi, 16 Graustufen, inkl. Software
DM 798,— zuzüglich DM 5,— Versandkosten

SCHLEGEL DATENTECHNIK
Schwarzachstr. 3 · 7940 Riedlingen · Tel. (07371) 2317

In der nächsten ST-Computer lesen Sie unter anderem

Die nächste Generation

- Motorola 68030 und Inmos T800

ATARIs Neuheiten der CeBIT setzen nicht nur im Grafikbereich neue Maßstäbe. Mit den verwendeten Prozessoren und der damit verbundenen Rechenleistung wird ein enorm günstiges Preis/Leistungsverhältnis geschaffen. Die Vorstellung der 80386-, 68030- und T800-Maschinen zeigt ganz deutlich den Willen des ATARI-Managements, sich im Semi-Professional und Professional-Bereich zu etablieren. Wir möchten Ihnen mit dem 68030 und dem T800 die beiden 32-Bit Prozessoren der neuen ATARI Systeme näherbringen.

DIN A4-Funktionsplotter

Programme zur Darstellung dreidimensionaler Funktionen gibt es in großer Menge - aber nur die wenigsten arbeiten mit einer Auflösung von 2000*1440 Punkte. Das vorgestellte Programm hat nicht nur diese Eigenschaft, es vermittelt zudem eine plastische Vorstellung der darzustellenden Funktion durch die Berücksichtigung der Helligkeitsverteilung auf der Oberfläche infolge einer (fiktiven) Lichtquelle.

Und ....

der für diese Ausgabe versprochene Artikel Laservisionen mit dem neuen Laserdruckertreiber für den ATARI SLM804. Ganz kurz noch einmal worüber es sich hierbei handelt:
Wer bis jetzt mit dem ATARI Laserdrucker SLM 804 gearbeitet hat, mußte leider sehr schnell gewisse Mängel an der Treibersoftware feststellen. Um dieses Manko zu beheben, veröffentlichen wir in zwei Teilen ab der nächsten Ausgabe einen neuen Laserdruckertreiber, mit dem sowohl Grafik, als auch Text problemlos auszudrucken sind. Alles über die Programmierung des Lasers und einen Grafiktreiber kann man in der Juni-Ausgabe finden.

Änderungen vorbehalten !

Die ST-Computer Ausgabe 6/88 erscheint am 27.5.1988

Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen kostet sehr viel Zeit und Mühe. Da wir ja weiterhin vorhaben, die Qualität zu steigern (ja, auch wenn das manchmal daneben geht), haben wir Redakteure ein großes Anliegen an Sie, liebe Leserinnen und Leser:

Bitte haben Sie Verständnis dafür, daß Fragen an die Redaktion in Zukunft nur noch zu bestimmten Zeiten beantwortet werden können. Wir stehen Ihnen zu folgenden Terminen telefonisch zur Verfügung:

Donnerstag von 14 - 17 Uhr

Vielen Dank für Ihr Verständnis

Ihre Redaktion

Impressum

ST Computer

Chefredakteur:
Uwe Bärtels (UB)

Redaktion:
Uwe Bärtels (UB)
Harald Egel (HE)
Marcelo Merino (MM)
Harald Schneider (HS)

Redaktionelle Mitarbeiter:
Claus Brod (CB)
Stefan Höhn (SH)
Raymund Hofmann (RH)
Oliver Joppich (OJO)
H.P.Labude (HP)
Jürgen Leonhard (JL)
Claus P. Lippert (CPL)
Markus Nerding (MN)
Chr. Schormann (CS)
Andreas Suchy (AS)
Jörg Wilhelm (JW)

Autoren dieser Ausgabe:
R.Bager
D.Brockhaus
I.Brümmer
A.Esser
M.Fangmeyer
P.Fischer
U.Gohlke
U.Meumann
R.Peiler (RP)
L.Reinirkens
Dr.K.Sarnow
M.Schuhmacher (MS)
R.Tolksdorf

Public Relations:
Claus P. Lippert (Leitung)
D.dela Fuente (UK)
L.Hennelly (Nordamerika)

Redaktion: "Merlin" Computer GmbH
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 11
FAX : 0 61 96/4 11 37

Verlag:
Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57
FAX : 0 61 51/5 56 89 + 5 60 59

Verlagsleitung:
H.J.Heim

Anzeigenverkaufsleitung:
U.Heim

Anzeigenverkauf:
K.Magaritis

Anzeigenpreise:
nach Preisliste Nr.3, gültig ab 1.1.88
ISSN 0932-0385

Grafische Gestaltung:
Fabian & Mayer

Fotografie:
R.Spirandelli, Archiv

Produktion:
K.H.Hoffmann, B.Failer, S.Failer

Druck:
Ferling Druck

Lektorat:
M.Menzenbach

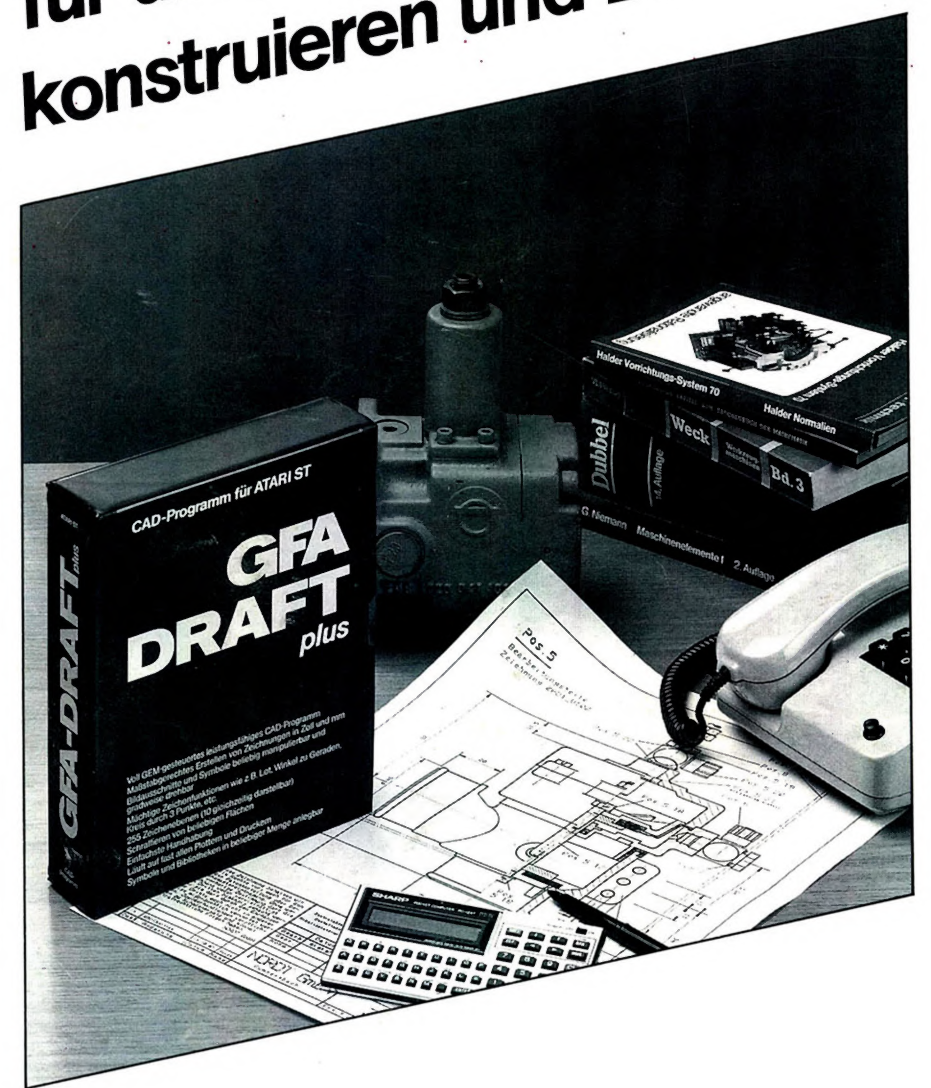
Bezugsmöglichkeiten:
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

ST Computer erscheint 11 x im Jahr
Einzelpreis: DM 7,-, ÖS 56,-, SFr 7,-
Jahresabonnement: DM 70,- Europ. Ausland : DM 90,-
Luftpost : DM 120,-

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern dem Heim Verlag.
Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

GFA-DRAFT plus für ST

das leistungsfähige CAD-Programm für alle, die entwerfen, konstruieren und zeichnen.



- Voll GEM-gesteuertes leistungsfähiges CAD-Programm
- Maßstabgerechtes Erstellen von Zeichnungen in Zoll und mm
- Bildausschnitte und Symbole beliebig manipulierbar und gradweise drehbar
- Mächtige Zeichenfunktionen wie z. B. Lot, Winkel zu Geraden, Kreis durch 3 Punkte, etc.
- 255 Zeichenebenen je bis DIN A 0 (10 gleichzeitig darstellbar)
- Schraffieren und Bemaßen von beliebigen Flächen
- Eingebaute Kommandosprache
- Einfachste Handhabung
- Läuft auf fast allen Plottern und Druckern
- Symbole und Bibliotheken in beliebiger Menge anlegbar
- Stücklisten-Verwaltung ist im Lieferumfang enthalten

DM 349,–

...Anruf genügt.

GFA-CLUB
GFA-ST/PC-Software
bitte Info anfordern

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 02 11/58 80 11

